Unsupervised Representative Selection and Signal Unmixing

by

Dung Ngoc Tran

A dissertation submitted to The Johns Hopkins University

in conformity with the requirements for the degree of

Doctor of Philosophy

Baltimore, Maryland

February, 2019

Abstract

This thesis presents unsupervised machine learning algorithms to tackle two related problems: selecting representatives in a dataset and identifying constituent components in mixture data. In both problems, we aim to reveal a few key hidden features that sufficiently explain the data. The main intuition behind our algorithms is that, in an appropriately constructed dictionary, a sparse representation of the data corresponds to selecting these unknown features. Our goal is to efficiently seek such sparse representations under suitable conditions.

In the representative selection problem, our objective is to pick a few representative data points that capture distinguished characteristics of a dataset. This corresponds to identifying the vertices of the polytope generated by the data. To do so, we start by modeling each data point as a convex combination of the polytope vertices. Then, in the dictionary formed by the dataset itself, we look for sparse representations of the data which subsequently imply the vertices. To seek such sparse representations, we proposed a greedy pursuit algorithm and a non-convex entropy minimization algorithm. We theoretically justify our proposed algorithms and demonstrate their vertex recovery performance on both synthetic and real data.

In the unmixing problem, we assume that each data point is a mixture of a few unknown components, and we wish to decompose data into these underlying constituents. We consider a highly under-sampled regime in which the number of measurements is far less than the data dimension. Furthermore, we solve an even more challenging unmixing problem in which the under-sampled mixture are indirectly observed via a nonlinear operator such as Sigmoid and Relu. To find the unknown constituents, we form a dictionaries with atoms resembling the constituents and seek the sparse representations corresponding to them. We proposed a fast and robust greedy algorithm, called UnmixMP, to find such sparse representations. We prove its robust unmixing performance and support our theoretical analysis by various experiments on both synthetic and real image data.

Our algorithms are fast and robust, and supported by rigorous theoretical analysis. Our experimental results shows that the proposed are significantly more robust than state-of-the-art representative selection and unmixing algorithms in the aforementioned settings.

Thesis Committee

Primary Readers

Trac D. Tran (Primary Advisor)
Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Sang P. Chin (Co-advisor)
Professor
Department of Computer Science
Boston University

Mark A. Foster
Associate Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Vishal M. Patel
Assistant Professor
Department of Electrical and Computer Engineering
Johns Hopkins Whiting School of Engineering

Dedication

To my parents, my wife and my daughter for their endless love and support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recent advances in deep learning have demonstrated that the curse of big data might in fact be a blessing in the supervised setting: large carefully-labeled training samples combined with massive computational resources lead to numerous breakthroughs from speech recognition to object/pattern classification to chess playing. Unfortunately, most real-world data are unlabeled or poorly labeled. Furthermore, real data are often incomplete and corrupted by various sources of interference. Therefore, unsupervised learning from raw data is one of the most important challenges in machine learning.

Real-world data are inherently sparse in certain domains in the sense that they can be approximately characterized by only a few significant informative components. In other words, the intrinsic signal information usually contains in specific well-defined low-dimensional structures. These underlying structures not only help gain insights into the data, hence assist decision making, but can also be incorporated into supervised machine learning algorithms which allows improvement in the performance of learning and inference tasks. Discovering these hidden structures from raw data belongs to the set of most

important and interesting machine learning problems in the unsupervised setting.

This thesis presents unsupervised machine learning algorithms to solve two related problems of learning underlying structures from data: representative selection and data unmixing. The former aims to identify a few data representatives that capture the most relevant information of a dataset. The latter focuses on extracting a few underlying components constituting the data mixture. These two type of data structures are ubiquitous in real-world applications.

1.1 Representative selection

In various applications, data are often redundant in the physical space. In other words, there is a small subset of the data, called *representatives* or *exemplars*, that appropriately represents the whole dataset. These exemplars can capture the underlying distribution of the data so that the same performances of inference algorithms as being applied to the original dataset can be archived at much lower costs. Additionally, they can themselves reveal hidden information of the data such as topics in text documents, endmembers in hyperspectral images, or key frames in the video summarization problem. The *representative selection* problem, sometimes called the *subset selection* problem, concerns choosing a few representative data points in a dataset.

The advantage of solving this problem is two-fold. On one hand, working directly with a small amount of data greatly improve memory and computational complexity efficiency. On the other hand, informative and interpretable

representatives can reveal hidden information and thus gain insights into the data and assist decision making. Fig. 1.1 demonstrates a text summarization example in which representative selection solutions offer physical contexts which lack from traditional word-count feature learning technique such as Principle Component Analysis (PCA) and Dictionary Learning (DL) (Elad, 2010).

Representative selection methods rely on the assumption that there is a small subset of data that sufficiently explains the entire dataset. This assumption is justified in various real applications. For example, in the topic modeling problem, there is often some document belonging to a certain topic, and one can infer the topics of a corpus from a small subset of these single-topic documents.

Two questions naturally arise in this problem: (1) how one defines representativeness, and (2) how to efficiently find the exemplars given that definition of representativeness. They lead to the corresponding challenges in the representative selection problem: (1) criteria that allows choosing informative and interpretable representatives and (2) fast and robust algorithms with theoretical guarantees to optimize the criteria. Several subset selection or representative selection criteria have been studied in the literature, including maximum cut objective (Kulesza and Taskar, 2012; Motwani and Raghavan, 1995), maximum marginal relevance (Carbonell and Goldstein, 1998), capacitated and uncapacitated facility location objectives (Mirchandani and Francis, 1990; G. L. Nemhauser and Fisher, 1978), multi-linear coding (Elhamifar, Sapiro, and Vidal, 2012; Esser et al., 2012) and maximum volume subset

Figure 1.1: *Advantage of representative selection over word-counting techniques such as Principle Component Analysis (PCA) and Dictionary Learning (DL).* **Top**: word-cloud representation of the reviews of an Amazon product. The size of a word corresponding the its frequency in the reviews: prominent words appear more frequently in the reviews. **Bottom**: A exemplar negative review (1-star) obtained by our proposed representative selection algorithm applied to the reviews of this product. The representatives offer physical context which is missing from the word-counting based techniques. For example, in the word-cloud, the word "burn" appears rather frequently which implies the product can possibly be used to burn discs. However, the representative offer indicates a situation in which the product failed to burn discs.

(Kulesza and Taskar, 2012; Borodin and Olshanski, 2000), which all try to characterize the informativeness/value of a subset of items in terms of ability to represent the entire distribution and/or having minimum information overlap

among selected items. On the other hand, optimizing almost all representative selection criteria is, in general, NP-hard and non-convex (Motwani and Raghavan, 1995; Feige, 1998; Gonzalez, 1985; Civril and Magdon-Ismail, 2009), which has motivated the development and study of approximate methods for optimizing these criteria. This includes greedy approximate algorithms (G. L. Nemhauser and Fisher, 1978) for maximizing submodular functions, such as graph-cuts and facility location, which have worst-case approximation guarantees, as well as sampling methods from Determinantal Point Process (DPP) (Kulesza and Taskar, 2012; Borodin and Olshanski, 2000), a probability measure on the set of all subsets of a ground set, for approximately finding the maximum volume subset. Motivated by the maturity of convex optimization and advances in sparse and low-rank recovery, recent methods have focused on convex relaxation-based methods for subset selection (Elhamifar, Sapiro, and Vidal, 2012; P. Awasthi and Ward, 2015; Nellore and Ward, 2015).

In our work, we aim to pick representatives that explain uniquely distinctive features of the data. For example, in the video summarization problem, we are interested in choosing a few key dark and light scenes in a video. Another example is online review summarization in which we look for a few positive and negative product reviews each of which demonstrates a unique property of the product. These representative reviews help users quickly and easily access the product quality.

1.2 Unmixing

Various types of data, though complex in the ambient space, have low intrinsic degree of freedoms. One particularly important structure is that each data sample is a mixture of a few constituent components. This typically arises when data are captured under the presence of different data sources. For instance, audio data acquired during a conference meeting are often superpositions of voices from different speakers. Alternatively, image data can consist of a background scene and different foreground objects. Separating underlying constituents from mixture data is the subject of the unmixing problem.

Various unmixing problems have been long studied in research areas spanning signal processing, statistics, and physics. One of the most basic challenges is that the unmixing problem is generally ill-posed. In particular, this problem suffers from a fundamental identifiability issue in which the number of observations is typically less than the number of unknowns. For example, in a simple case of unmixing a speech sample recorded from two speakers talking at the same time, each observation is the superposition of two samples, each from a different source. Identifying individual voice samples in this case is thus equivalent to solving for two unknowns given each observation.

Furthermore, unmixing poses an additional challenge of undersampling. This arises in scenarios such as occlusions or missing data. This causes the measurement number typically much less than the data dimension. This poses a great challenge to the unmixing problem.

To overcome the aforementioned challenges, further structural assumptions on the constituent signals are necessary, and have been the focus of significant research over the last few years. For the unmixing problem to have an identifiable solution, one typically assumes some form of *incoherence* between the constituent components (Elad et al., 2005; Donoho et al., 2006). In particular, the underlying components are assumed to be sufficiently "distinct" so that the recovery problem is well-posed. Furthermore, to deal with the undersampling issue, the degree of freedom of the constituents are assumed to be small. This is the notion of *sparsity*.

In the simplest setting, consider the linear model:

$$\boldsymbol{z} = \boldsymbol{\Phi x} + \boldsymbol{\Psi y}. \tag{1.1}$$

Here, $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are called dictionaries. Each of their columns contains an elementary structure that might appear in the corresponding constituent elements. The coefficient vector $\boldsymbol{x}$ selects the columns of $\boldsymbol{\Phi}$ that appear in the first constituent, while $\boldsymbol{x}$ selects the columns of $\boldsymbol{\Psi}$ that generate the second signal. Incoherence dictates that the columns of the dictionaries are weakly correlated, and sparsity requires that the coefficient vectors have few nonzero elements.

In the undersampling setting,

$$\boldsymbol{y} = \boldsymbol{A}(\boldsymbol{\Phi x} + \boldsymbol{\Psi y}). \tag{1.2}$$

where $\boldsymbol{A}$ is the undersampling operator that might account for known occlusions or missing data.

The unmixing problems under the above linear models have been studied significantly over the past few years. The work of (Starck et al., 2003) uses (1.1) to model the problem of distinguishing stars from galaxies in an astronomical image. The work in (Elad et al., 2005; J. Bobin and Donoho, 2007) posed the unmixing problem as an instance of morphological components analysis (MCA), and formalized the observation model (1.2). Specifically, these approaches posed the recovery problem in terms of a convex optimization procedure, such as the LASSO (Tibshirani, 1996). The work of Pope et al. (C. Studer and ÌĺLolcskei, 2012) analyzed somewhat more general conditions under which stable unmixing could be achieved. More recently, the work of (McCoy and Tropp, 2014) showed a curious phase transition behavior in the performance of the convex optimization methods. Specifically, they demonstrated a sharp statistical characterization of the achievable and non-achievable parameters for which successful unmixing of the signal components can be achieved. Moreover, they extended the unmixing problem to a large variety of signal structures beyond sparsity via the use of general atomic norms in place of the ℓ_1-norm. See (M. McCoy and Baldassarre, 2014) for an in-depth discussion of atomic norms, their statistical and geometric properties, and their applications to unmixing.

In this work, we consider an even more challenge unmixing problem in which the linear undersampled measurements are indirectly observed via a nonlinear operator. Furthermore, the observations can be corrupted by dense additive noise.

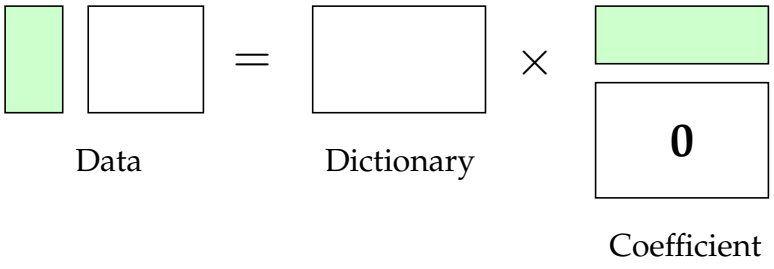


Figure 1.2: *Representative selection via sparse representation*. The nonzero rows (colored box) of the coefficient identify to the representatives (colored block in the data)

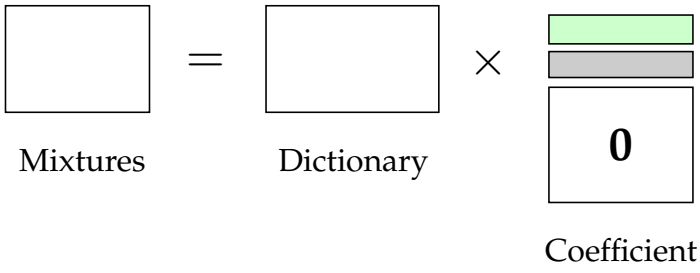


Figure 1.3: *Unmixing via sparse representation*. The nonzero rows (colored block) of the coefficient can be used to reconstruct the underlying constituents. Here, different colored blocks correspond to different sources.

1.3 Main idea: sparse representation

The main intuition underlying our approaches is that the degree of freedom in both representative selection and signal unmixing problems is typically low. In the former case, the number of representatives are often much smaller than the total number of data points, while there are only a few constituent components underlying each high dimensional mixture signal. We can therefore identifying these features by seeking a sparse representation of the data points in an appropriate dictionary. Fig 1.2 and Fig 1.3 illustrates this idea.

A sparse representation not only leads to faster processing algorithms

but also more effective signal separation as it focuses on the most relevant property of the data. Additionally, sparse representations allow us to capture hidden simplified structures in the data, and thus minimizes the harmful effects of noisy in practical settings.

Last but not least, sparsity models and algorithms have been under significant research in the past 20 years, with applications widely range from neuroscience, computational biology, and computer vision. It is thus beneficial to utilize insights and results from this line of research.

1.4 Representative selection meets data unmixing

Real-world applications justify that the representative selection problem can sometimes be considered as a special case of an unmixing problem. This can be seen in the hyperspectral unmixing or endmember extraction problem as depicted in Fig1.4. In hyperspectral imagery, each pixel typically consists of a mixture of the reflectance spectra of several materials where the mixture coefficients correspond to the abundances of the constituent materials. In the hyperspectral unmixing problem, one aims to extract from an input hyperspectral image the original spectral signals of some set of constituting materials. Intuitively, this unmixing problem can be solved by seeking a sparse representation of the hyperspectral data in a dictionary whose atoms resemble the spectral reflectance patterns of the underlying materials. Unfortunately, this information is generally unavailable a priori. Recent advances in nonnegative matrix factorization and hyperspectral imaging justify the assumption that

there are often some pure pixels for each underlying materials in a hyperspectral image (Arora et al., 2012; Bittorf et al., 2012; Gillis and Luce, 2014; Qu et al., 2014; Tran et al., 2015). That is, each of these pure pixels contains the spectral signal of a certain single material, and the unmixing problem reduces to identifying these pure pixels from the input data. Furthermore, each mixture pixel can be approximately represented by a few pure pixels. Therefore, one can extract the pure pixels by finding a joint sparse representation of all pixels in the dictionary formed by the pixels themselves.

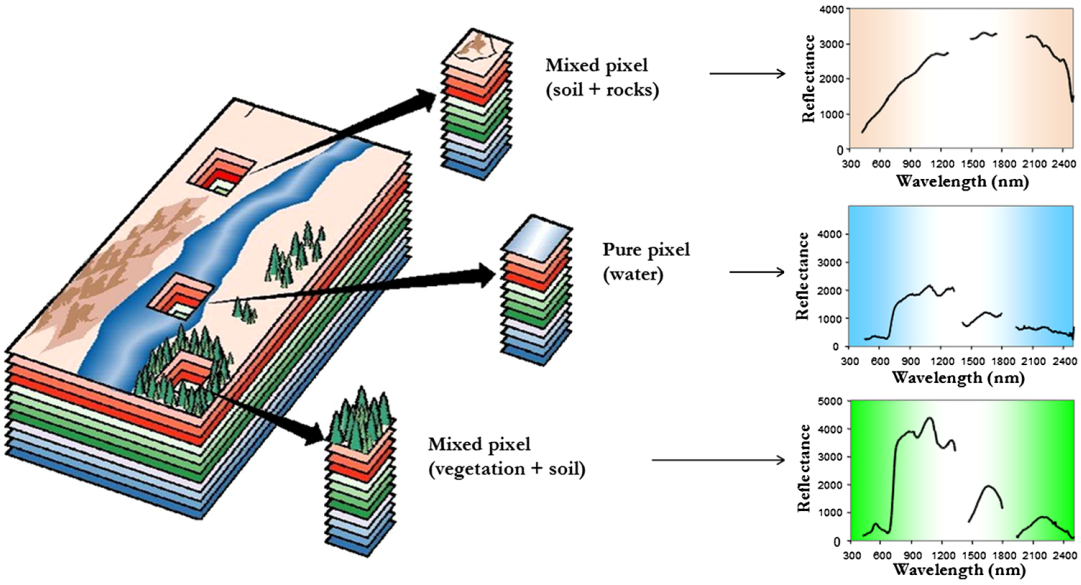


Figure 1.4: *Data unmixing reduces to representative selection.* Each pixel of a hyperspectral image is typically a mixture of the reflectance spectra of several materials. The hyperspectral unmixing problem aims to extract the original spectral signals of some set of prime materials, which can reduce to identifying a set of pure pixels or representatives. Each of these pure pixels contains the spectral signal of a single material, and others pixels can be represented as mixtures of these pure pixels.

1.5 Thesis contributions

We propose fast and provably correct algorithms for tackling the representative selection and unmixing problems.

1.5.1 Greedy algorithms and non-convex models for representative selection

In our work, we aim to pick representatives that explain uniquely distinctive characteristics of the data. For example, given several thousands of text reviews of an Amazon product, our goal is to select a few positive reviews and negative reviews of the product that allows fast decision making. Each of these representative reviews indicates a certain unique property of the product. Another application is in the video summarization problem in which we aim to choose a few key frames of the video that allows users to quickly evaluate the video content. For instance, in a news video, the chosen representatives should include a few frames showing the presenter, a few frames showing the reported event, and so on.

More specifically, we consider the convex hull, or polytope, generated by the data points. Fig 1.5 illustrates a toy example of a data polytope. Our goal is to fast and reliably identify the data columns corresponding to the vertices, the purple color points in Fig 1.5, of this polytope. Empirical evidences indicate that these vertices indeed captures unique characteristics of the data, and can well present the other data points. Under this model, we propose two fast and robust representative selection algorithms and prove their correctness. In particular, we introduce (1) a greedy pursuit algorithm that iteratively picks

the vertices based on a carefully chosen criterion, and (2) a non-convex model based on entropy minimization which concentrates some form of energy on the vertices.

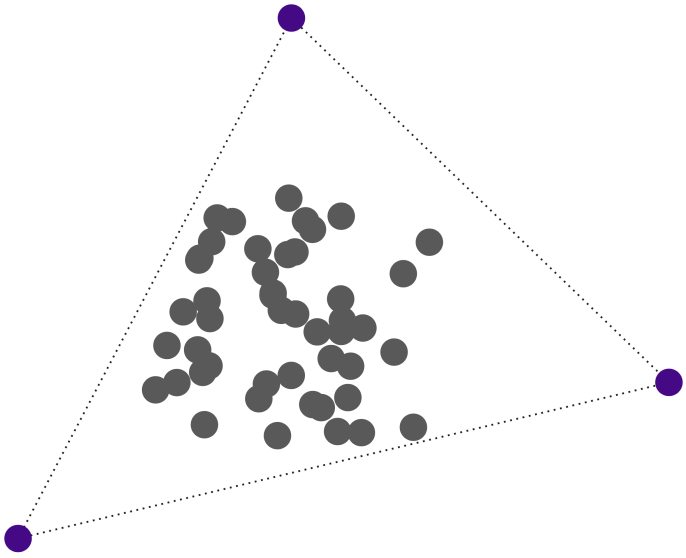


Figure 1.5: *Data convex hull toy example.* The purple points are the the vertices of the data convex hull, and are chosen as representatives by the proposed representative selection algorithms.

The key idea behind the proposed algorithms is that, under the convex hull model, the entire dataset can be represented by the representatives via a simple mathematical relationship. Consequently, in the dictionary formed by the dataset itself, a joint sparse representation of the data points corresponds to selecting the vertices. Our goal is to find such sparse representations under the aforementioned relationship between the dataset and the representatives.

Finding such sparse representations can be formulated as a combinatoric optimization problem which is, in general, NP-hard (Motwani and Raghavan,

1995). Our proposed algorithms are essentially greedy approximation and non-convex relaxation methods to efficiently solve such combinatoric optimization programs.

1.5.1.1 Gradient vertex pursuit

To seek such sparse representations under the convex hull model, we first propose a greedy pursuit algorithm, called Gradient Vertext Pursuit (GVP), that iteratively choose the vertices until the entire data convex hull is identified.

More specifically, throughout the GVP algorithm, we maintain an estimate of the convex hulls and incrementally augment this set one vertex at each iteration. Each vertex is chosen to guarantee that the convex hull estimate at the corresponding iteration best approximates the true convex hull, dictated by some loss function. We prove that after a finite steps, where the number of steps equals to the number of the vertices, our GVP algorithm is guaranteed to correctly identify the convex hull. We further empirically show that GVP not only possesses the flexibility and low complexity of a typical greedy algorithm (and thereby faster than linear and convex optimization methods) but is also more robust than other greedy pursuit algorithms for solving the convex hull problem.

1.5.1.2 Non-convex entropy minimization

We next propose a non-convex relaxation to the aforementioned combinatoric optimization problem that seeks the sparse representation for identifying the representatives. Specifically, we introduce a row sparsity measure based on the entropy function over the shared sparse representations of the data. We show

rigorously that by minimizing this measure under the convex hull assumption, one can robustly recover the vertices even when the data is corrupted by noise. We call the resulting optimization problem Row Entropy Minimization (REM).

As we will show in the experiment section, solving REM leads to better solutions comparing to state-of-the-art convex hull algorithms. The trade-off of is that the row entropy objective in REM is nonconvex due to the non-convexity of the entropy function. We thus approximate the objective function by its first order approximation and utilize an iterative algorithm to solve a series of simple convex subproblems.

1.5.2 Greedy algorithms for separating signals from nonlinear compressive observations

We tackle a challenging unmixing problem in which the linear undersampled measurements are indirectly observed via a nonlinear operator and possibly corrupted by dense additive noise.

In particular, we propose a fast and robust iterative algorithm called *UnmixMP* to unmix component signals from nonlinear compressive mixtures. At a high level, in each iteration of the algorithm consists of two main step. First, it aims to identify a true dictionary atom for each component signal. As we show later in the corresponding chapter, each such atom is most correlated with the gradient of the loss function that is evaluated at the component signal estimated from the previously identified atoms. Second, we finer estimate each constituent signal based on those chosen atoms and all corresponding dictionary atoms previously selected. Our algorithm belongs to the class of

greedy pursuit algorithms which has been receive lots of attention in sparse recovery literature (Mallat and Zhang, 1993; Tropp and Gilbert, 2007; Dai and Milenkovic, 2009; Needell and Tropp, 2009).

The algorithm enjoys an attractive common property of greedy pursuit algorithm that it requires no annoying parameter tuning. In its standard form, UnmixMP only requires the sparsity level of each component vectors. This information is often available from domain specific knowledge. Even when it is unavailable, one can declare successful recovery by stopping the algorithm when the reconstruction error falls below a certain small threshold. This insight is supported by our theoretical result on the upper bound of the iteration number the corresponding chapter.

We rigorously show that UnmixMP is fast and robust. In particular, for certain observation models, we prove that the reconstruction errors of the unmixed signals decay linearly. Unlike other convex and thresholding unmixing methods (Soltani and Hegde, 2016), this property comes at no cost of parameter tuning in the main loop of the algorithm. Furthermore, we also prove that the sample complexity to achieve this linear convergence rate is upper bounded by $\mathcal{O}\left(r \log \frac{N}{r}\right)$, where r is the total sparsity level of the component signals.

In addition, we support our theoretical analysis by various experiments on both synthetic and real image data. We demonstrate that our algorithm is significantly more robust than state-of-the-art unmixing algorithms in this nonlinear setting.

Last but not least, each step of UnmixMP identifies atoms from constituent

dictionaries separately. This allows parallelized implementation to speed up the algorithm. Detailed discussion on this subject will be presented in the corresponding chapter.

1.6 Thesis outline

The rest of the thesis is outlined as follows. Chapter 2 reviews some background materials that are used throughout the thesis.In Chapter 3, we present our unsupervised learning algorithm for the representative selection problem. We pose this problem as a sparse recovery problem, and derive our proposed algorithms. We study the theoretical guarantees of the proposed algorithms, and demonstrate their robust representative selecting performance on both synthetic as well as the real-world problems of hyperspectral unmixing, video summarization, and text summarization. In Chapter 4, study the problem of unmixing nonlinearly observed mixture data under ill-sampled regime. We present our proposed greedy pursuit algorithm for tackling this problem and theoretically justify its robust unmixing performance. We support our theoretical results with supporting empirical results on both synthetic and real image data. Finally, we summarize our work in Chapter 5.

References

Elad, M. (2010). "Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing". In: *Appl. Comput. Harmonic Analysis*.

Kulesza, A. and B. Taskar (2012). "Determinantal point processes for machine learning". In: *Foundations and Trends in Machine Learning* 5.

Motwani, R. and P. Raghavan (1995). "Randomized algorithms". In: *Cambridge University Press*.

Carbonell, J. and J. Goldstein (1998). "The use of mmr, diversity-based reranking for reordering documents and producing summaries". In: *SIGIR*.

Mirchandani, P. B. and R. L. Francis (1990). "Discrete Location Theory". In:

G. L. Nemhauser, L. A. Wolsey and M. L. Fisher (1978). "An analysis of approximations for maximizing submodular set functions". In: *Mathematical Programming* 14.

Elhamifar, E., G. Sapiro, and R. Vidal (2012). "See all by looking at a few: Sparse modeling for finding representative objects". In: *IEEE Conference on Computer Vision and Pattern Recognition*.

Esser, E., M. Moller, S. Osher, G. Sapiro, and J. Xin (2012). "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space". In: *IEEE Transactions on Image Processing* 21, pp. 3239–3252.

Borodin, A. and G. Olshanski (2000). "Distributions on partitions, point processes, and the hypergeometric kernel". In: *Communications in Mathematical Physics* 211.

Feige, U. (1998). "A threshold of ln n for approximating set cover". In: *Journal of the ACM*.

Gonzalez, T. (1985). "Clustering to minimize the maximum intercluster distance". In: *Theoretical Computer Science* 38.

Civril, A. and M. Magdon-Ismail (2009). "On selecting a maximum volume sub-matrix of a matrix and related problems". In: *Theoretical Computer Science* 410.

P. Awasthi A. S. Bandeira, M. Charikar R. Krishnaswamy S. Villar and R. Ward (2015). "Relax, no need to round: Integrality of clustering formulations". In: *Conference on Innovations in Theoretical Computer Science (ITCS)*.

Nellore, A. and R. Ward (2015). "Recovery guarantees for exemplar-based clustering". In: *Information and Computation*.

Elad, M., J. Starck, P. Querre, and D. Donoho (2005). "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)". In: *Appl. Comput. Harmonic Analysis* 19.3, pp. 340–358.

Donoho, D., M. Elad, and V. Temlyakov (2006). "Stable recovery of sparse overcomplete representations in the presence of noise". In: *IEEE Trans. Inform. Theory* 52.1, pp. 6–18.

Starck, J.-L., D. L. Donoho, and E. J. CandÃĺs (2003). "Astronomical image representation by the curvelet transform". In: *Astronom. Astrophys.* 392.2, pp. 785–800.

J. Bobin J. Starck, J. Fadili Y. Moudden and D. Donoho (2007). "Morphological component analysis: An adaptive thresholding strategy". In: *IEEE Trans. Image Proc.* 16.11, 2675âĂŞ2681.

Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso". In: *J. Royal Statist. Soc B* 58.1, 267âĂŞ288.

C. Studer P. Kuppinger, G. Pope and H. B ÌĹolcskei (2012). "Recovery of sparsely corrupted signals". In: *IEEE Trans. Inform. Theory* 58.5, 3115âĂŞ3130.

McCoy, M. and J. Tropp (2014). "Sharp recovery bounds for convex demixing, with applications". In: *Foundations of Comp. Math.* 14.3, 503âĂŞ567.

M. McCoy V. Cevher, Q. Dinh A. Asaei and L. Baldassarre (2014). "Convexity in source separation: Models, geometry, and algorithms". In: *IEEE Sig. Proc. Mag.* 31.3, 87âĂŞ95.

Arora, S., R. Ge, R. Kannan, and A. Moitra (2012). "Computing a nonnegative matrix factorization: provably". In: *Proceedings of the 44th symposium on Theory of Computing*, pp. 145–162.

Bittorf, V., B. Recht, C. Re, and J.A. Tropp (2012). "Factoring nonnegative matrices with linear programs". In: *NIPS*, pp. 1223–1231.

Gillis, N. and R. Luce (2014). "Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization". In: *Journal of Machine Learning Research* 15, pp. 1249–1280.

Qu, Q., X. Sun, N.M. Nasrabadi, and T.D. Tran (2014). "Subspace vertex pursuit for separable non-negative matrix factorization in hyperspectral unmixing". In: *IEEE International Conference onAcoustics, Speech and Signal Processing (ICASSP)*, pp. 8115–8119.

Tran, Dung N., Tao Xiong, Sang Peter Chin, and Trac D. Tran (2015). "Nonnegative matrix factorization with gradient vertex pursuit". In: *ICASSP*.

Mallat, Stéphane G. and Zhifeng Zhang (1993). "Matching pursuits with time-frequency dictionaries". In: *IEEE Transactions on signal processing* 41.12, pp. 3397–3415.

Tropp, Joel A. and Anna C. Gilbert (2007). "Signal recovery from random measurements via orthogonal matching pursuit". In: *IEEE Transactions on information theory* 53.12, pp. 4655–4666.

Dai, Wei and Olgica Milenkovic (2009). "Subspace pursuit for compressive sensing signal reconstruction". In: *IEEE Transactions on Information Theory* 55.5, pp. 2230–2249.

Needell, Deanna and Joel A. Tropp (2009). "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". In: *Applied and Computational Harmonic Analysis* 26.3, pp. 301–321.

Soltani, Mohammadreza and Chinmay Hegde (2016). "Fast Algorithms for Demixing Sparse Signals from Nonlinear Observations". In: *arXiv preprint arXiv:1608.01234*.

Chapter 2

Background

We use bold uppercase letters for matrices, and bold lowercase letters for column vectors. The notion $\boldsymbol{x} \in \mathbb{R}^n$ denotes a column vector consisting of n real-valued elements, and $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ indicates a real-valued $n \times m$ matrix. We work in the usual Euclidean space $\mathbb{R}^n$ with the canonical basis $\{\boldsymbol{e}_j\}_{j=1}^n$. We define $\Omega_{\mathcal{S}}^n = \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{z} \geq \boldsymbol{0}, \|\boldsymbol{z}\|_1 = 1, \text{and } z_k = 0 \text{ for all } k \notin \mathcal{S}\}$ as the probability simplex in the subspace spanned by $\{\boldsymbol{e}_j\}_{j \in \mathcal{S}}$ in $\mathbb{R}^n$. The transpose operator is denoted by $[\cdot]^\mathsf{T}$.

The notations $\boldsymbol{0}_k$ and $\boldsymbol{1}_k$ denote the all-zero and all-one vectors of length k, respectively. We let $\boldsymbol{I}_k$ be the identity matrix in $\mathbb{R}^{k \times k}$. Without subscripts, the sizes of these vectors and matrices will be inferred from the context. Given a matrix $\boldsymbol{Y}$, we let $\boldsymbol{y}_i$, $\boldsymbol{y}^j$, and $y_{i,j}$ denote its i-th column, j-th row and (i, j) element, respectively. For an index set $\mathcal{S}$, the matrix $\boldsymbol{Y}_{\mathcal{S}}$ consists of the columns of $\boldsymbol{Y}$ whose indices supported by $\mathcal{S}$. For a set Γ, we use $|\Gamma|$ to denote its cardinality and Γ^c denote its complement. The notation $\mathbb{R}_+$ denotes nonnegative numbers. Similar notations are used for higher dimensional vector spaces.

2.1 Vector and matrix norms

Given a vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix} \in \mathbb{R}^n$, its ℓ_0 is defined as

$$\|\boldsymbol{x}\|_0 := \sum_{j=1}^{n} I_{\{z \in \mathbb{R} | z \neq 0\}}(x_j). \tag{2.1}$$

Here, $I(\cdot)$ is the indicator function given by

$$I_{\Omega}(z) = \begin{cases} 1 & : z \in \Omega, \\ 0 & : z \notin \Omega. \end{cases} \tag{2.2}$$

In other words, $\|\boldsymbol{x}\|_0$ counts the nonzero elements of $\boldsymbol{x}$.

The ℓ_p norm, for $p > 0$, is defined as

$$\|\boldsymbol{x}\|_p := \sum_{j=1}^{n} \left(|x_j|^p\right)^{1/p}. \tag{2.3}$$

Taking $p \to \infty$ results in the ℓ_∞ norm which can be shown to be equal to the maximum absolute values of the elements of its argument. That is,

$$\|\boldsymbol{x}\|_\infty = \max_j |x_j|. \tag{2.4}$$

Note that the ℓ_0 is not a real vector norm as it does not satisfy all the properties of a vector norm. It can therefore be referred as a pseudo-norm.

Consider a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$, its entry-wise matrix norms treat $\boldsymbol{X}$ as a vector of size nm and apply vector norms to this vector. More specifically, the $p-$norm of $\boldsymbol{X}$ is given by

$$\|\boldsymbol{X}\|_p = \|\mathrm{vec}(\boldsymbol{X})\|_p = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} |x_{ij}|^p \right)^{1/p}. \tag{2.5}$$

The special case $p = 2$ is the Frobenius norm, which is also denoted as $\|\cdot\|_F$,

and taking $p \to \infty$ results in the maximum norm.

The matrix mixed norm $\ell_{1,q}$, where $q > 0$, is the sum of the ℓ_q norm of the rows of its argument. In particular,

$$\|\boldsymbol{X}\|_{1,q} = \sum_{i=1}^{n} \|\boldsymbol{x}^i\|_q. \tag{2.6}$$

For example, $\|\boldsymbol{X}\|_{1,\infty}$ is the sum of the ℓ_∞ norm of the rows of $\boldsymbol{X}$.

The $\ell_{1,q}$ norm can be generalized to the $\ell_{p,q}$ norm, for $p, q > 0$, which is defined as

$$\|\boldsymbol{X}\|_{p,q} = \left(\sum_{i=1}^{n} \|\boldsymbol{x}^i\|_q^p \right)^{1/p}. \tag{2.7}$$

A special case of the matrix mixed norm is the $\ell_{\text{row},0}$ norm which is given by

$$\|\boldsymbol{X}\|_{\text{row},0} = \sum_{i=1}^{n} I_{\{\boldsymbol{z} \in \mathbb{R}^m | \boldsymbol{z} \neq \boldsymbol{0}\}}(\boldsymbol{x}^j). \tag{2.8}$$

In other words, it counts the number of nonzero rows of its argument. It can also be regarded as a generalization of the vector ℓ_0 norm.

2.2 Sparse representation

In this section, we review some of the key concepts in the the sparse representation theory, which we use throughout the thesis.

2.2.1 Sparse coding

A vector $\boldsymbol{x} \in \mathbb{R}^d$ is called an s-sparse vector, for $s \leq m$, if it has at most s nonzero coefficients. Given an observed signal $\boldsymbol{y} \in \mathbb{R}^n$, we seek to approximately represent $\boldsymbol{y}$ as an s-sparse vector $\boldsymbol{x}_* \in \mathbb{R}^d$ with respect to a dictionary matrix $\boldsymbol{D}$ of size $n \times d$. This problem can be solved via the following ℓ_0-minimization program (Cand'es and Tao, 2005; Cand'es, Romberg, and Tao, 2006; Cand'es and Romberg, 2006; Cand'es and Tao, 2006; Donoho, 2006)

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}. \tag{2.9}$$

In the noisy setting, one solves

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 \leq \epsilon, \tag{2.10}$$

for $\epsilon > 0$.

As the ℓ_0-minimization program is NP-hard, the following ℓ_1- minimization (Cand'es and Tao, 2005; Cand'es, Romberg, and Tao, 2006; Cand'es and Romberg, 2006; Cand'es and Tao, 2006; Donoho, 2006) is proposed to as a convex relaxation of the ℓ_0-minimization

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \boldsymbol{y} = \boldsymbol{D}\boldsymbol{x}, \tag{2.11}$$

and

$$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{s.t.} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2 \leq \epsilon, \tag{2.12}$$

in the noisy setting, where $\epsilon > 0$.

2.2.2 Joint Sparse coding

We call a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$ an $s-$row-sparse matrix if it has at most s nonzero rows. Given a data matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$, we seek to approximately represent all data columns of $\boldsymbol{Y}$ by the same subset of a dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{n\times d}$. This problem can be cast as solving the following row sparse minimization program which finds the smallest number of dictionary atoms that represent the data (Tropp, C., and J., 2006b; Tropp, C., and J., 2006a)

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{\text{row},0} \quad \text{s.t.} \quad \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}. \tag{2.13}$$

In the noisy setting, we instead solve

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{\text{row},0} \quad \text{s.t.} \quad \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F \leq \epsilon, \tag{2.14}$$

For some $\epsilon > 0$.

This row sparse optimization problem is intractable and NP-hard. One thus can instead solve the following convex relaxation of the row sparse minimization problem (Tropp, C., and J., 2006b; Tropp, C., and J., 2006a)

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{1,q} \quad \text{s.t.} \quad \boldsymbol{Y} = \boldsymbol{D}\boldsymbol{X}, \tag{2.15}$$

or

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{1,q} \quad \text{s.t.} \quad \|\boldsymbol{Y} - \boldsymbol{D}\boldsymbol{X}\|_F \leq \epsilon, \tag{2.16}$$

for noisy data, where $\epsilon > 0$.

References

Cand'es, E. J. and T. Tao (2005). "Decoding by linear programming". In: *IEEE Trans. on Information Theory* 51.12, pp. 4203–4215.

Cand'es, E. J., J. Romberg, and T. Tao (2006). "Robust uncertainty principles: exact signal re- construction from highly incomplete frequency information". In: *IEEE Trans. on Information Theory* 52, pp. 489–509.

Cand'es, E. J. and J. Romberg (2006). "Quantitative robust uncertainty principles and optimally sparse decompositions". In: *Foundations of Computational Mathematics* 6, pp. 227–254.

Cand'es, E. J. and T. Tao (2006). "Near optimal signal recovery from random projections: universal encoding strategies?" In: *IEEE Trans. on Information Theory* 52.12, pp. 5406–5425.

Donoho, D. (2006). "Compressed sensing". In: *IEEE Trans. on Information Theory* 52.4, pp. 1289–1306.

Tropp, Joel A., Gilbert A. C., and Strauss M. J. (2006b). "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit". In: *Signal Processing* 86.3, pp. 572–588.

Tropp, Joel A., Gilbert A. C., and Strauss M. J. (2006a). "Algorithms for simultaneous sparse approximation. Part I: Convex relaxation". In: *Signal Processing* 86.3, pp. 589–602.

Chapter 3

Sparse modeling and algorithms for representative selection

In this chapter, we consider the problem of choosing a few representative data points from a dataset. Our objective is to choose such representatives, or exemplars, that describe distinguished features of the data, such as featured positive and negative reviews of an Amazon product or typical bright and dark scenes in a movie. We propose two algorithms, namely Gradient Vertex Pursuit (GVP) and Row Entropy Minimization (REM), to tackle this problem. The key idea behind the proposed algorithms is that, under appropriate assumptions, the entire dataset can be characterized by the representatives via a simple mathematical relationship. Consequently, in the dictionary formed by the dataset itself, a joint sparse representation of the data points corresponds to selecting the exemplars. The proposed algorithms are fast and efficiently find such sparse representations under the aforementioned relationship between the dataset and the representatives.

The advantage of choosing distinguished representatives from a dataset is two-fold. On one hand, working directly with a small amount of data greatly

improve memory and computational complexity efficiency. On the other hand, informative and interpretable representatives can reveal hidden information and thus gain insights into the data and assist decision making.

In the subsequent sections, we first state precisely the mathematical model used to choose exemplars. Next, we review existing methods for choosing representatives based on such model. We then introduce our proposed algorithms. We theoretically justify their correctness in recovering representatives under the aforementioned assumption, and empirically demonstrate their efficacy on both synthetic and real datasets.

3.1 Problem formulation

Consider a collection of m noise-free data points $\mathcal{Y} = \{\boldsymbol{y}_j\}_{j=1}^{m} \subset \mathbb{R}^n$ in Fig 3.1. Let

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_1 & \dots & \boldsymbol{y}_m \end{bmatrix} \tag{3.1}$$

denote the data matrix whose columns are the data points. The representative selection problem refers to the problem of finding a small subset of data points that sufficient characterizes the dataset given a certain criterion.

In this work, we aim to pick representatives that not only explain the data, preferably via a simple mathematical model, but also characterize distinguished features of the dataset. We argue that the vertices of the polytope generated by a dataset offer these desirable properties. Our goal is then to pick these unknown polytope vertices as the data representatives. We state our representative-picking criterion in the following assumption.

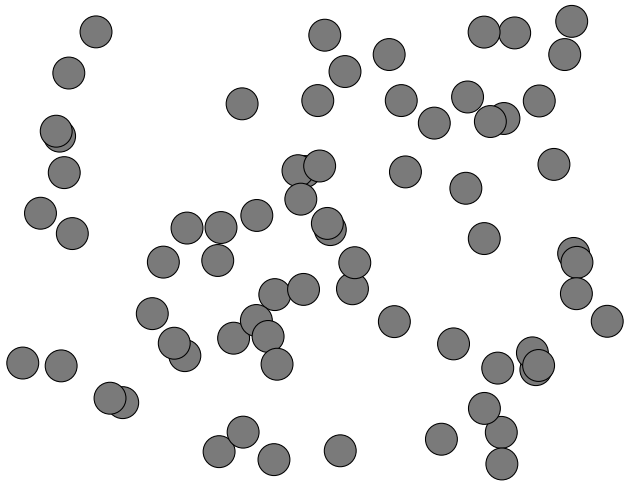


Figure 3.1: *A set of m noise-free data points in* $\mathbb{R}^n$

Assumption 1 (Convex hull assumption)**.** *Given a matrix* $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$*, there exists an index set* $\mathcal{S}$ *of cardinality s, for some positive integer* $s < \min\{n, m\}$*, such that* $\boldsymbol{Y} = \boldsymbol{Y}_{\mathcal{S}}\boldsymbol{X}$ *where* $\boldsymbol{X} \in \mathbb{R}_{+}^{s\times m}$ *satisfying* $\mathbf{1}^T\boldsymbol{X} = \mathbf{1}^T$*.*

The columns of $\boldsymbol{Y}_{\mathcal{S}}$ are called vertices of the dataset, and we choose them as the data representatives. Our representative selection problem can then be stated as:

Given the data matrix $\boldsymbol{Y}$ ***satisfying Assumption 1, find the vertex index set*** $\mathcal{S}$***.***

In the next sections, we present proposed algorithms to solve this problem. In the rest of this section, we justify Assumption 1, and argue that choosing data vertices as representatives is justified in various real applications.

To begin, the following definitions provide some initial insights.

Definition 2. *The convex hull of a finite point set* $\mathcal{Y} = \{\boldsymbol{y}_j\}_{j=1}^m$ *is the set of all*

convex combinations of its points:

$$conv\,(\mathcal{Y}) = \left\{ \sum_{j=1}^{m} x_j \boldsymbol{y}_j \;\middle|\; x_j \geq 0, \forall j,\ and\ \sum_{j=1}^{m} x_j = 1 \right\}. \tag{3.2}$$

Definition 3. *The convex hull conv* $(\mathcal{Y})$ *of a finite set of data points* $\mathcal{Y}$ *forms a convex polytope. Each* $\boldsymbol{y} \in \mathcal{Y}$ *for which* $\boldsymbol{y} \notin conv\,(\mathcal{Y} \setminus \boldsymbol{y})$ *is called a vertex of conv* $(\mathcal{Y})$*. A vertex of conv* $(\mathcal{Y})$ *is also called an extreme point of* $\mathcal{Y}$*. We denote the index set of the vertices as* $\mathcal{S}$*, and the vertex set as* $\mathcal{Y}_{\mathcal{S}}$*.*

The summation in (3.2) is called a *convex combination* of the data points in $\mathcal{Y}$. The above definitions allow us to nicely express the dataset as a simple matrix factorization. In particular, they indicate that the finite set of data points $\mathcal{Y}$ generates a polytope, each point of which can be can be represented as a convex combination of all data points. We call it the data polytope. Furthermore, the data points themselves can be expressed via such convex combinations. This can be written in a matrix form as

$$\boldsymbol{Y} = \boldsymbol{Y}\boldsymbol{X}, \tag{3.3}$$

where $\boldsymbol{X} \in \mathbb{R}_{+}^{m \times m}$ and $\mathbf{1}^{\mathsf{T}}\boldsymbol{X} = \mathbf{1}^{\mathsf{T}}$. In this equation, each column of $\boldsymbol{X}$ is the set of coefficients in a convex combination representing the corresponding data column, e.g., $\boldsymbol{y}_k = \sum_{j=1}^{m} x_{j,k}\boldsymbol{y}_j, \forall k$, where $x_{j,k} \geq 0, \forall j,$ and $\sum_{j=1}^{m} x_{j,k} = 1$.

The set of all coefficient matrix $\boldsymbol{X}$ that satisfies (3.3) is nonempty. To see this, note that the identity matrix is trivially a solution of this equation. In this case, we consider each data point as a convex combination of itself. This means any given finite dataset can be expressed as (3.3).

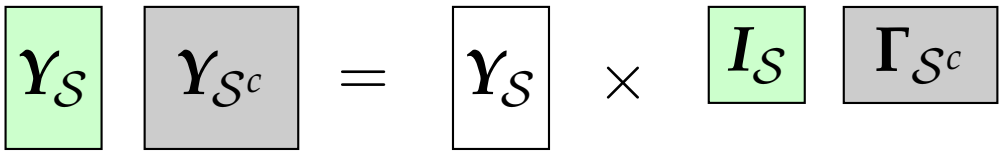


Figure 3.2: *Illustration of equation (3.5).*

Definition (3) offers an interesting insight. It suggests that each vertex of the data polytope can only be represented as a convex combination of itself. Furthermore, the following proposition shows that the set of vertices of the data polytope can fully characterize the entire dataset.

Proposition 4. *Given a finite dataset $\mathcal{Y}$ and its corresponding vertex set $\mathcal{Y}_\mathcal{S}$,*

$$conv\left(\mathcal{Y}\right) = conv\left(\mathcal{Y}_\mathcal{S}\right). \tag{3.4}$$

In other words, any data point can be expressed as a convex combination of the vertices. That is, for each data point $\boldsymbol{y}_k$, there is a set of coefficients $\{x_{j,k}\}_{j=1}^{s}$ satisfying $x_{j,k} \geq 0, \forall j \in \mathcal{S}$, and $\sum_{j=1}^{m} x_{j,k} = 1$, such that $\boldsymbol{y}_k = \sum_{j=1}^{s} x_{j,k}\boldsymbol{y}_j$. Here, $s = |\mathcal{S}|$ is the number of vertices. This allows us to rewritten (3.3) as:

$$\boldsymbol{Y} = \boldsymbol{Y}_\mathcal{S}\boldsymbol{X}, \tag{3.5}$$

with $\boldsymbol{X} \in \mathbb{R}_{+}^{s\times m}$ satisfying $\mathbf{1}^T\boldsymbol{X} = \mathbf{1}^T$. Note that unlike in (3.3) where the coefficient matrix has m rows corresponding to all data points, the coefficient matrix $\boldsymbol{X}$ in this equation has s rows which matches number of vertices. Fig 3.2 visualizes this equation.

In short, the vertices of the data polytope posses desirable properties that

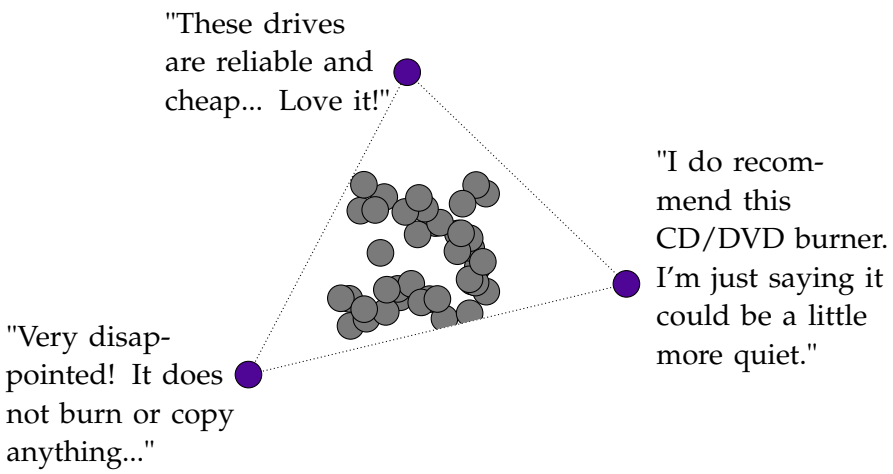


Figure 3.3: *Illustration of choosing vertices as representatives.*

are suitable for our representative-picking criterion. That is, the entire dataset can be explained by the vertices via (3.5). Furthermore, each vertex can only be expressed as a convex combination of itself. This means each vertex possesses a unique feature in the dataset. This is depicted in Fig 3.3

This assumption is justified in several applications such as text modeling, hyperspectral unmixing, and blind source separation (Arora et al., 2009; Chan et al., 2008; Bioucas-Dias et al., 2012; Kumar, Sindhwani, and Kambadur, 2012; Tran et al., 2015). Throughout the paper, we assume that the vertices of this convex hull are *distinct*.

3.2 Sparse representation for selecting representatives

The equation in Assumption 1 can be rewritten as

$$\boldsymbol{Y} = \boldsymbol{Y}\boldsymbol{X}, \tag{3.6}$$

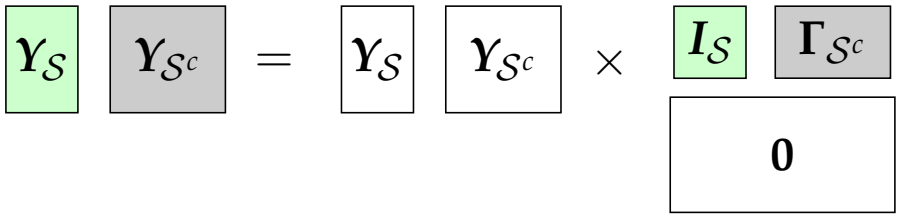


Figure 3.4: *Rewritten Equation (3.5)*

where $\boldsymbol{X} \in \mathbb{R}^{m \times m}$ such that each column of $\boldsymbol{X}$ sums to one, and at most s rows of $\boldsymbol{X}$ are nonzero. These rows of $\boldsymbol{X}$ are supported by $\mathcal{S}$, meaning they corresponds to the vertices. Then picking the representatives or vertices under this assumption becomes finding the s nonzero rows of $\boldsymbol{X}$ satisfying these above constraints. Fig 3.4 describes this process. In the language of sparse representation, the problem can be modeled as (Tran et al., 2015; Tran et al., 2016)

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{\text{row},0} \quad \text{s.t.} \quad \boldsymbol{Y}\boldsymbol{X} = \boldsymbol{Y},\ \boldsymbol{X} \geq \mathbf{0},\ \mathbf{1}^\mathsf{T}\boldsymbol{X} = \mathbf{1}^\mathsf{T}, \tag{3.7}$$

where $\|\boldsymbol{X}\|_{\text{row},0}$ counts the number of nonzero rows of $\boldsymbol{X}$. The *distinct* vertices of the dataset can then be identified by extracting the nonzero rows of an optimal solution returned by this row sparse problem. The following proposition justify this insight.

Proposition 5. *Given a dataset $\mathcal{Y}$ that satisfy the convex hull assumption 1. Let $\mathcal{S}$ be the vertex index set. Assume that the columns are distinct and $\boldsymbol{Y} = \begin{bmatrix}\boldsymbol{Y}_{\mathcal{S}} & \boldsymbol{Y}_{\mathcal{S}^c}\end{bmatrix}$. Let $\boldsymbol{X}_*$ be an optimal solution of the optimization problem (3.7), then*

$$\boldsymbol{X}_* = \begin{pmatrix} \boldsymbol{I}_s & \boldsymbol{\Gamma} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \in \mathbb{R}^{m \times m}, \tag{3.8}$$

for some $\boldsymbol{\Gamma} \geq \mathbf{0}$ and $\mathbf{1}^\mathsf{T}\boldsymbol{\Gamma} = \mathbf{1}^\mathsf{T}$.

When the data are perturbed by noise, the following robust versions of the above optimization program offer a robust approach for selecting representatives:

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{\text{row},0} \quad \text{s.t.} \quad \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) \leq \epsilon,\ \boldsymbol{X} \geq \boldsymbol{0},\ \mathbf{1}^{\mathsf{T}}\boldsymbol{X} = \mathbf{1}^{\mathsf{T}}, \tag{3.9}$$

where $\epsilon > 0$. Here $\mathcal{L}(\cdot)$ is a loss function that represents the consistency between the data and the factorization. Common choices of the loss function are the Frobenius norm

$$\mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) = \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}\|_F$$

and the L_1 norm

$$\mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) = \|\boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}\|_1 .$$

Choosing an appropriate loss function depends on the nature of the data. For example, the L_1 norm is suitable for data with sparse and large noise whereas the Frobenius norm is often chosen when there is dense and small noise.

The optimization problem (3.9) can be interpreted as choosing the smallest number of representatives that represent the data up to an error ϵ as convex combinations. In general, solving this robust version yields more reasonable results than the exact recovery problem (3.7). In real applications, there can be nearly duplicate data which possibly causes the exact row sparse problem to pick all several similar vertices or representatives.

Another optimization program, which is closely related to (3.9), is

$$\min_{\boldsymbol{X}} \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) \quad \text{s.t.} \quad \|\boldsymbol{X}\|_{\text{row},0} \leq k\ \boldsymbol{X} \geq \boldsymbol{0},\ \mathbf{1}^{\mathsf{T}}\boldsymbol{X} = \mathbf{1}^{\mathsf{T}}, \tag{3.10}$$

for some $k > 0$. Instead of focusing on the reconstruction error of the data

and letting the solution decides the number of representatives as in (3.9), this optimization problem seeks a pre-defined number of representatives that best describe the dataset in the convex combination sense. This program has a useful property that its optimal solutions not only specify the representatives but also provide the ranking information of the representatives. This can be seen by solving (3.10) for a small number of representatives, such as $k = 1$, and for many representatives, respectively. In the former case, the optimal solutions pick the most important or informative vertices. In the latter case, less informative vertices will be identified along with the most important ones. This can be seen more clearly in the relaxation programs the the row sparse problem. We will discuss this matter further in the Row Entropy Minimization section.

In short, identifying the vertices of a dataset can be done by solving the row sparse problem (3.7) and its robust variants. Unfortunately, they are NP-hard and intractable. In the next section, we review the most important algorithms to tackle this problem.

3.3 Previous work

The representative selection problem is in fact a special of the popular *Non-negative Matrix Factorization* (NMF) (Lee and Seung, 1999) problem. In the NMF problem, one aims to factorize an input matrix into the product of two simpler nonnegative matrices which reveals certain interesting structures of the input matrix. This problem, though finds itself in enormous number of applications in various fields, is ill-posed (Donoho and Stodden, 2003) and

NP-hard (Vavasis, 2009). Most traditional methods rely on solving a non-convex optimization problem which lack of optimality guarantee (Lee and Seung, 2000).

Recently, it has been shown that under the *separable* assumption, the NMF problem admits a unique solution (Donoho and Stodden, 2003).

Definition 6 (Separable NMF)**.** *A data matrix $\boldsymbol{Y}$ is s-separable if there exists a cone generated by a few columns of $\boldsymbol{Y}$ that contains the entire dataset.*

The representative selection problem under the convex hull constraint is a special case of the separable NMF problem in which we impose the sum-to-one constraint to the coefficients to obtain interpretability of the selected representatives.

Based on separable assumption, several elegant algorithms have been introduced in literature to solve the separable NMF problem. (Kumar, Sindhwani, and Kambadur, 2012; Arora et al., 2012; Bittorf et al., 2012; Kumar and Sindhwani, 2013; Gillis and Luce, 2014; Gillis and Vavasis, 2014; Gillis, 2014; Esser et al., 2012; Elhamifar, Sapiro, and Vidal, 2012) which aim to recover the set of extreme points by either solving easier linear programming or convex optimization problems, or by adopting a greedy pursuit procedure. In particular, the greedy pursuit approaches iteratively identify the extreme points based on geometric intuition. Due to its geometric nature, these greedy algorithms are typical less robust to noise (Kumar, Sindhwani, and Kambadur, 2012; Kumar and Sindhwani, 2013; Gillis and Vavasis, 2014; Gillis, 2014). The linear programming methods solve a linear programming under the separability constraints and aim to optimize the representation of the vertices only

(Gillis and Luce, 2014; Bittorf et al., 2012). They are thus also less robust to noise. The convex optimization method (Elhamifar, Sapiro, and Vidal, 2012; Esser et al., 2012) solves the row sparse minimization problem using a convex relaxation. They aim to optimize the representation of the entire dataset under the separability constraints and thus are robust to noise. However, they suffer from the duplicate data problem in which they emphasize duplicate or near-duplicate extreme points equally. This leads to identical or similar representatives in the output.

Our proposed algorithms in some sense solve the NMF problem under the convex hull constraint. As shown in the experimental result sections, they are robust to noise and able to deal with duplicate or near-duplicate data. Furthermore, we offer rigorous theoretical guarantees for both the algorithms.

3.4 Gradient vertex pursuit

Throughout this section, we assume that the input data matrix $\boldsymbol{Y}$ satisfies Assumption 1, and let $(\boldsymbol{Y}; \mathcal{S}, \mathcal{C})$ denote the data, its vertex index set, and the corresponding data convex hull. We define $\Omega^n_{\mathcal{S}} = \{\boldsymbol{z} \in \mathbb{R}^n : \boldsymbol{z} \geq \boldsymbol{0}, \|\boldsymbol{z}\|_1 = 1, \text{and } z_k = 0 \text{ for all } k \notin \mathcal{S}\}$ as the probability simplex in the subspace spanned by $\{\boldsymbol{e}_j\}_{j \in \mathcal{S}}$ in $\mathbb{R}^n$.

We consider a class of loss functions satisfying the separability.

Assumption 7. *The loss function $\mathcal{L}$ is separable into the sum of functions of the individual columns of its argument; i.e., $\mathcal{L}(A) = \sum_j f(A_j)$ where $f : \mathbb{R}^n \to \mathbb{R}_+$ is a strongly convex function with parameter $\mu > 0$ and its gradient is Lipschitz continuous with constant L. That is, $L\boldsymbol{I} \succ \nabla^2 f \succ \mu \boldsymbol{I}$ where $\nabla^2 f$ is the Hessian of*

f. We further assume that $f(0) = 0$ if and only if $x = 0$.

Therefore, (3.10) is equivalent to

$$\min \sum_j f(\boldsymbol{y}_j, \boldsymbol{Y}\boldsymbol{x}_j) \;\; \text{s.t.} \;\; \|\boldsymbol{X}\|_{0,\text{row}} \leq s, \;\; \boldsymbol{X} \geq \boldsymbol{0}, \;\; \boldsymbol{1}^T\boldsymbol{X} = \boldsymbol{1}^T. \tag{3.11}$$

Here, by abuse of notation, we use $f(\boldsymbol{y}_j, \boldsymbol{Y}\boldsymbol{x})$ as a function of $\boldsymbol{x}$ that represents the consistency between $\boldsymbol{y}_j$ and $\boldsymbol{Y}\boldsymbol{x}$. Similar notation can be inferred from the context.

The strongly convexity of f allows fast implementation of the algorithm. To solve (3.10), we introduce a fast, robust, and provably correct greedy pursuit algorithm, *Gradient Vertex Pursuit* (GVP), based on these assumptions. As we will show in the experiment section, GVP, while possessing the flexibility and low complexity of a typical greedy algorithm (and thereby faster than linear and convex optimization methods), is more robust than other greedy pursuit algorithms for solving the representative selection problem under Assumption 1.

Our solution relies on a greedy approach: given the input data $(\boldsymbol{Y}; \mathcal{S}, \mathcal{C})$, we maintain an estimate of $\mathcal{S}$ and incrementally augment this set one vertex at an iteration. This estimate at some iteration t is denoted by $\mathcal{S}^t$ and the convex hull generated by $\{\boldsymbol{y}_j\}_{j \in \mathcal{S}^t}$ is represented by $\mathcal{C}^t$. Furthermore, we call $\mathcal{C}^t$ the *sub-polytope* of $\mathcal{C}$ at iteration t. The basic intuition of our method stems from the following observation.

Claim 8. *If there is a point lying outside a sub–polytope $\mathcal{C}^t$ of the original vertex hull, there exists at least one vertex which does not belong to $\mathcal{C}^t$.*

Proof. If the point in the hypothesis is a vertex, Claim 8 holds trivially. Otherwise, if all vertices are in $\mathcal{C}^t$, there exists at least one point exterior to the original convex hull which leads to a contradiction. □

Finding such a point is easy and can be done efficiently by solving the convex optimization problem:

$$\min \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}_{\mathcal{S}^t}\boldsymbol{X}) \quad \text{s.t.} \quad \boldsymbol{X} \geq \boldsymbol{0},\ \boldsymbol{1}^T\boldsymbol{X} = \boldsymbol{1}^T. \tag{3.12}$$

It can be easily seen that solving (3.12) is equivalent to solving

$$\min \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) \quad \text{s.t.} \quad \boldsymbol{x}_j \in \Omega^m_{\mathcal{S}^t}, \forall j \in \{1, ..., m\}. \tag{3.13}$$

Let $\boldsymbol{X}^t$ be the optimal solution to (3.13); each column of the matrix $\boldsymbol{Y}\boldsymbol{X}^t$ is the projection of the corresponding data point onto $\mathcal{C}^t$. Consequently, the zero columns of the residual matrix $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}^t$ correspond to the interior points of the sub–polytope, whereas the residuals of the data points lying outside $\mathcal{C}^t$ are nonzero.

The main concern now is on a strategy for vertex identification given a sub–polytope and some exterior point. The following lemma suggests a way to proceed.

Lemma 9. *Let $\boldsymbol{y}_l$ be a column lying outside a sub–polytope $\mathcal{C}^t$, and let $\boldsymbol{X}^t$ be the optimal solution to (3.13), then*

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} \geq \min_{k \in \mathcal{S}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_k} \tag{3.14}$$

for any $j \in \{1, ..., m\}$.

This lemma can be proved by applying the chain rule to the left hand side of (3.14) and then utilizing Assumption 1.

Proof. For any $j \in \{1, ..., m\}$, applying the chain rule to the left hand side of (3.14) yields

$$\begin{aligned}\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_j} &= \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\frac{\partial(\boldsymbol{Y}\boldsymbol{x})}{\partial x_j} \\ &= \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\frac{\partial(x_j\boldsymbol{y}_j)}{\partial x_j} \\ &= \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\boldsymbol{y}_j. \end{aligned} \tag{3.15}$$

As $\boldsymbol{Y}$ satisfies Assumption 1, there is a coefficient vector $\bar{\boldsymbol{x}} \in \Omega_{\mathcal{S}}^m$ such that

$$\boldsymbol{y}_j = \boldsymbol{Y}\bar{\boldsymbol{x}} = \sum_{k\in\mathcal{S}} \bar{x}_k \boldsymbol{y}_k. \tag{3.16}$$

Thus, (3.15) becomes

$$\begin{aligned}\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_j} &= \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\sum_{k\in\mathcal{S}} \bar{x}_k \boldsymbol{y}_k \\ &= \sum_{k\in\mathcal{S}} \bar{x}_k \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\boldsymbol{y}_k. \end{aligned} \tag{3.17}$$

Following the derivation step of (3.15), we have

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial(\boldsymbol{Y}\boldsymbol{x})}\boldsymbol{y}_k = \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_k}.$$

Plugging this into (3.17) yields

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_j} = \sum_{k\in\mathcal{S}} \bar{x}_k \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_k}.$$

As $\bar{\boldsymbol{x}} \in \Omega_{\mathcal{S}}^{m}$, we have that $\bar{\boldsymbol{x}} \geq \mathbf{0}$. Therefore, the above equality implies

$$\begin{aligned}\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_j} &\geq \sum_{k \in \mathcal{S}} \bar{x}_k \min_{k \in \mathcal{S}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_k} \\ &= \left(\min_{k \in \mathcal{S}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_k} \right) \sum_{k \in \mathcal{S}} \bar{x}_k. \end{aligned} \tag{3.18}$$

Again, it follows from the fact $\bar{\boldsymbol{x}} \in \Omega_{\mathcal{S}}^{m}$ that $\sum_{k \in \mathcal{S}} \bar{x}_k = 1$, we conclude that

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_j} = \left(\min_{k \in \mathcal{S}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x})}{\partial x_k} \right)$$

which completes the proof for Lemma 9. □

Lemma 9 is a generalization of a basic result in polyhedra theory: there exists at least one vertex that is an optimal solution to the problem of maximizing a linear function over a polytope (Cook et al., 1998). Indeed, if f is chosen to be the l_2 loss and $\boldsymbol{R} = \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}^t$ is the residual matrix at iteration t, then (3.14) becomes

$$\boldsymbol{R}_l^T \boldsymbol{y}_j \leq \max_{k \in \mathcal{S}} \boldsymbol{R}_l^T \boldsymbol{y}_k \tag{3.19}$$

for all $j \in \{1, ..., m\}$. As a result of the lemma, a vertex can be identified by minimizing the left hand side of (3.14) over the whole data set. In fact, this is the *greedy selection criteria* that we will use in the algorithm. Importantly, it can be proved that none of the vertices of the sub–polytope is an optimal solution to this minimization problem.

Lemma 10. *Assuming $\boldsymbol{y}_l$ is a column lying outside a sub–polytope $\mathcal{C}^t$ with $\boldsymbol{X}^t$ as the*

optimal solution to (3.13), the following holds:

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_l} < \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_k} \tag{3.20}$$

for any $k \in \mathcal{S}^t$.

The proof for this lemma starts from the optimality condition of (3.13). It can easily be obtained by combining a few simple facts and assumptions: $\boldsymbol{e}_k \in \Omega^n_{\mathcal{S}^t}$ for any $k \in \mathcal{S}^t$, f is convex, $f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{e}_l) = 0$, and $f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t) > 0$. A column whose index uniquely minimizes the left hand side of (3.14) is in fact a vertex that does not belong to $\mathcal{C}^t$.

Proof. To begin, notice that as $\boldsymbol{X}^t$ is the optimal solution to (3.13), it follows from the optimality condition of (3.13) that

$$\nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}(\boldsymbol{z} - \boldsymbol{x}_l^t) \geq 0,$$

for all $\boldsymbol{z} \in \Omega^m_{\mathcal{S}^t}$.

Next, for any $k \in \mathcal{S}^t$, it holds trivially that $\boldsymbol{e}_k \in \Omega^m_{\mathcal{S}^t}$. The above optimality condition thus implies

$$\nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}\boldsymbol{x}_l^t \leq \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}\boldsymbol{e}_k. \tag{3.21}$$

Now, it follows from the convexity of f that

$$f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t) + \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}(\boldsymbol{e}_l - \boldsymbol{x}_l^t) \leq f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{e}_l).$$

Equivalently,

$$f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t) - f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{e}_l) + \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}\boldsymbol{e}_l \leq \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T}\boldsymbol{x}_l^t. \tag{3.22}$$

As $\boldsymbol{Y}\boldsymbol{e}_l = \boldsymbol{y}_l$, it follows that

$$f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{e}_l) = f(\boldsymbol{y}_l, \boldsymbol{y}_l) = 0$$

by Assumption 7.

Furthermore, as $\boldsymbol{y}_l$ lies outside $\mathcal{C}^t$, it holds thus $\boldsymbol{y}_l \neq \boldsymbol{Y}\boldsymbol{x}_l^t$. This implies

$$f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t) > 0$$

by Assumption 7.

Therefore, it can be inferred from (3.22) that

$$\nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T} \boldsymbol{e}_l < \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T} \boldsymbol{x}_l^t. \tag{3.23}$$

Combining (3.21) and (3.23), we conclude that

$$\nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T} \boldsymbol{e}_l < \nabla f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)^\mathsf{T} \boldsymbol{e}_k,$$

or equivalently,

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_l} < \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_k}$$

which completes the proof for the lemma. □

With these lemmas, we now conclude this section with our main theorem.

Theorem 11. *Given a sub-polytope $\mathcal{C}^t$ and an exterior point $\boldsymbol{y}_l$, and let $\boldsymbol{X}^t$ be the optimal solution to (3.13); if the optimization problem*

$$\min_{j \in \{1,\ldots,m\}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} \tag{3.24}$$

has a unique solution k^{t+1}, then $k^{t+1} \in \mathcal{S} \setminus \mathcal{S}^t$.

Proof. It suffices to show

$$\min_{j\in\{1,\dots,m\}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} = \min_{j\in\mathcal{S}-\mathcal{S}^t} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j}$$

First, notice that since $\boldsymbol{y}_l \notin \mathcal{C}_{\mathcal{S}^t}$, we have $l \in \{1,...,m\} - \mathcal{S}^t$. It thus can be deduced from lemma 9 that

$$\min_{j\in\{1,\dots,m\}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} = \min_{j\in\{1,\dots,m\}-\mathcal{S}^t} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} \tag{3.25}$$

Furthermore, it follows from lemma 10 that

$$\frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_k} \geq \min_{j\in\mathcal{S}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j}$$

for any $k \in \{1,...,m\} - \mathcal{S}^t$. As a result,

$$\min_{j\in\{1,\dots,m\}-\mathcal{S}^t} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j} = \min_{j\in\mathcal{S}-\mathcal{S}^t} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j}$$

which together with (3.25) complete the proof for the theorem.

□

The following algorithm naturally follows from the analysis in the previous section.

Algorithm 12 (GVP).

Input: *matrix* $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ *satisfying Assumption 1, the number of vertices* s.

Output: *A set* $\tilde{\mathcal{S}}$ *of cardinality* s*, and a matrix* $\tilde{\boldsymbol{X}} \in \mathbb{R}^{m\times m}$.

Procedure:

1. *Initialize the vertex set estimate* $\mathcal{S}^0 = \emptyset$*, the coefficient* $\boldsymbol{X}^0 = 0$*, the residual* $\boldsymbol{R}^0 = 0$*, and the iteration counter* $t = 0$.

2. *Arbitrarily choose a nonzero residual* R_l^t*, and find*

$$k^{t+1} = \underset{j \in \{1,\dots,m\}}{\operatorname{argmin}} \frac{\partial f(\boldsymbol{y}_l, \boldsymbol{Y}\boldsymbol{x}_l^t)}{\partial x_j}$$

3. *Augment* $\mathcal{S}^{t+1} = \mathcal{S}^t \cup \{k^{t+1}\}$.

4. *Project* $\boldsymbol{Y}$ *onto* $\mathcal{C}^{t+1}$ *by finding* $\boldsymbol{X}^{t+1}$ *that solves:*

$$\min \mathcal{L}(\boldsymbol{Y}, \boldsymbol{Y}\boldsymbol{X}) \quad s.t. \quad \boldsymbol{x}_j \in \Omega^m_{\mathcal{S}^{t+1}}, \forall j \in \{1,\dots,m\},$$

   where $\Omega^m_{\mathcal{S}^{t+1}} = \{\boldsymbol{z} \in \mathbb{R}^m : \boldsymbol{z} \geq \boldsymbol{0}, \|\boldsymbol{z}\|_1 = 1, \text{and } z_k = 0 \text{ for all } k \notin \mathcal{S}^{t+1}\}$.

5. *Compute the residual* $\boldsymbol{R}^{t+1} = \boldsymbol{Y} - \boldsymbol{Y}\boldsymbol{X}^{t+1}$.

6. *Set* $t = t + 1$*; return to step 2 if* $t < s$*, otherwise, terminate the algorithm.*

7. *Set* $\tilde{\mathcal{S}} = \mathcal{S}^t$ *and* $\tilde{\boldsymbol{X}} = \boldsymbol{X}^t$.

Fig 3.5 Step 2 of the algorithm formalizes the greedy selection criteria mentioned in the previous section. The intuition behind it can be seen by letting f be the l_2 loss function. At initialization, this step finds the column with the largest l_2 norm. Moreover, at next iterations, it identifies the column that most correlates to the chosen nonzero residual. As a result of Theorem 11, one of the vertices is selected at each iteration and none can be ever chosen twice. This result is stated in Theorem 13 whose proof can be obtained easily by applying Theorem 11.

Theorem 13 (Correctness of GVP)**.** *If the input data* $(\boldsymbol{Y}; \mathcal{S}, \mathcal{C})$ *satisfies Assumption 1 and the optimization problem at step 2 at each iteration has a unique solution, the GVP algorithm correctly identifies all vertices after exactly* $|\mathcal{S}|$ *iterations.*

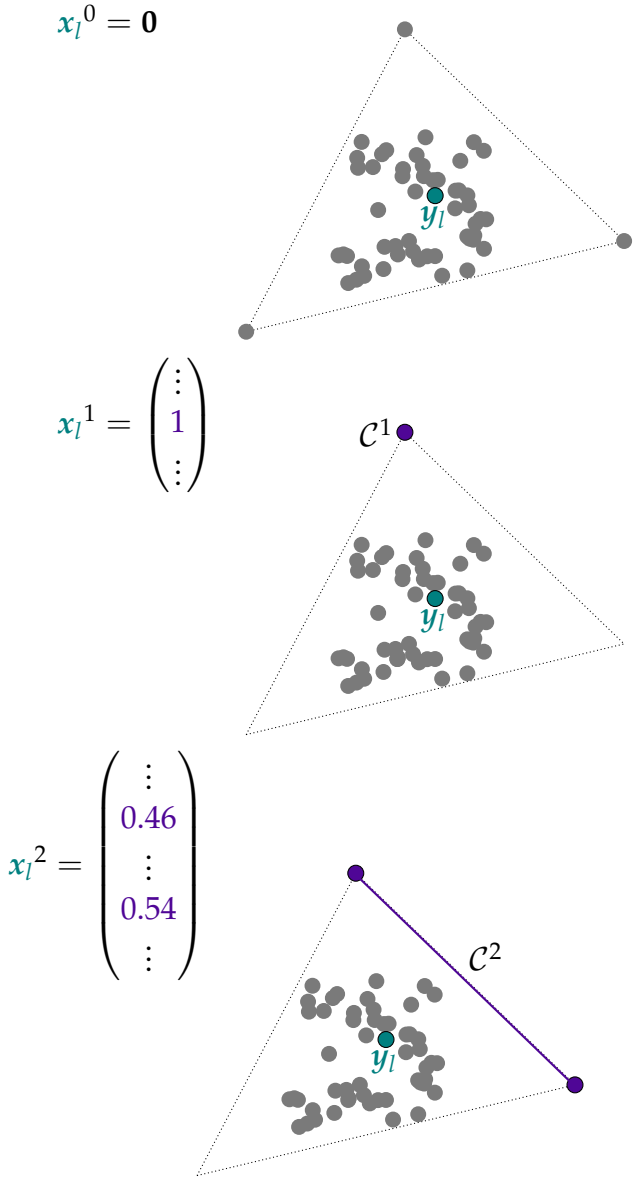


Figure 3.5: *Illustration of GVP.*

The solution of the optimization problem at step 4 in iteration t represents the coefficient of the data matrix when all data is projected onto the corresponding sub–polytope. This minimization problem is convex, thus can be solved efficiently by many off-the-shelf optimization solvers.

Remark. Similar to (Kumar, Sindhwani, and Kambadur, 2012), the following action can be performed to deal with the situation when (3.24) has multiple different optimal solutions at certain iteration t. If the solutions are vertices, all of them are added to $\mathcal{S}^t$; otherwise, GVP can be called recursively to identify the vertices of this set of solutions and add them to $\mathcal{S}^t$. Strikingly, the unique solution assumption in Theorem 13 is not strict: it satisfies with a high probability when data is randomly distributed. The proof for this claim is nontrivial, thus beyond the scope of the paper.

Moreover, the performance of the algorithm is not affected by the presence of duplicate columns. If step 2 results in identical vertices, only one of them is added to the vertex set estimate. As shown in Theorem 11, none of them is selected during subsequent iterations.

It is important to note that our method is different from the XRAY algorithm (Kumar, Sindhwani, and Kambadur, 2012) (Kumar and Sindhwani, 2013) as the latter fails to solve the representative selection problem under the convex hull assumption. This can be seen by considering a counter example shown in Figure 3.6a. Furthermore, in many cases, although XRAY successfully identifies the polytope vertices, it fails when there is small perturbation in the data as shown in Figure 3.6b.

Computational complexity. The GVP algorithm requires $\mathcal{O}(nms)$ operations in total. A comparison of its complexity to several state-of-the-art algorithms is shown in Table 3.1. Here, c is the number of iterations performed in the ADMM algorithm for solving the ℓ_{12}-minimization problem (Elhamifar,

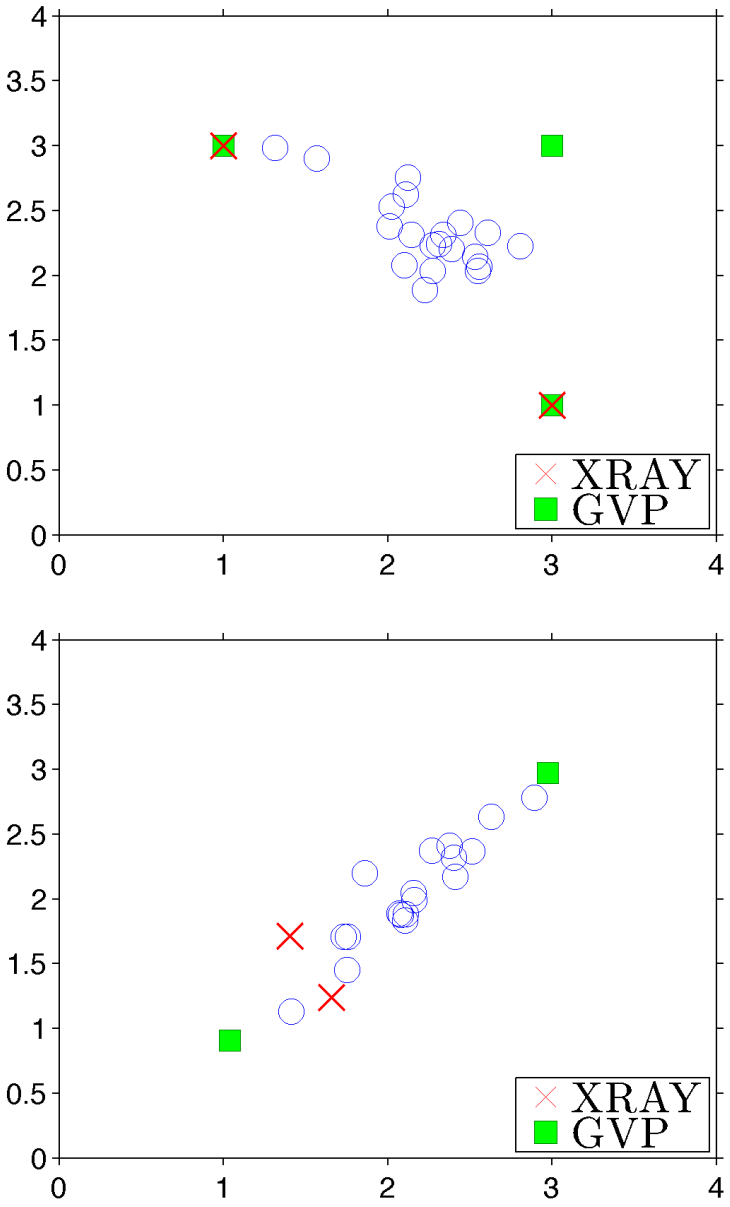


Figure 3.6: *A counter example*. **Top**: Data contained in a triangle with vertices $(3, 1)$, $(1, 3)$, and $(3, 3)$. **Bottom**: Noisy version of a dataset distributed on the line connecting vertices $(1.1, 1)$ and $(3, 2.9)$. XRAY fails in both cases, whereas GVP correctly identifies all vertices.

Sapiro, and Vidal, 2012).

Table 3.1: *Computational complexity comparison.*

VCA (Nascimento and Dias, 2004)	$\mathcal{O}(nms)$
SPA (Gillis and Vavasis, 2014)	$\mathcal{O}(nms)$
XRAY (Kumar, Sindhwani, and Kambadur, 2012)	$\mathcal{O}(nms)$
GVP	$\mathcal{O}(nms)$
$\ell_{1,2}$ (Elhamifar, Sapiro, and Vidal, 2012)	$\mathcal{O}(cm^3)$

3.4.1 Numerical experiments

This section evaluates the GVP algorithm on both synthetic and real data, and compare its performance to those of various greedy algorithms in Table 3.1. We also compare our algorithm to the $\ell_{1,2}$–minimization method (Elhamifar, Sapiro, and Vidal, 2012) to show that GVP, while being greedy, has almost the same superior performance as this convex relaxation method. We use a similar experiment setup to (Qu et al., 2014) and let f to be the ℓ_2 loss in all experiments.

3.4.1.1 Synthetic data

We test the robustness of our proposed algorithm against noise on a USGS library [1]. For each simulation, the data is generated as follows. Each column of the vertex matrix $\boldsymbol{Y}_{\mathcal{S}} \in \mathbb{R}^{n\times s}$ is randomly selected from the library; the coefficient matrix $\boldsymbol{X} \in \mathbb{R}^{s\times m}$ has the form of $\Gamma[\boldsymbol{I}_s, \boldsymbol{X}']$, where $\boldsymbol{I}_s \in \mathbb{R}^{s\times s}$ is the identity matrix, each column of $\boldsymbol{X}' \in \mathbb{R}_+^{m\times(m-s)}$ follows from a Dirichlet distribution whose parameters are chosen from a uniform distribution on $[0, 1]$ and Γ is a permutation matrix. The data matrix is generated by $\boldsymbol{Y} = \boldsymbol{Y}_{\mathcal{S}}\boldsymbol{X} + N$

[1] http://www.lx.it.pt/ bioucas/code/sunsal demo.zip

Table 3.2: *Running time comparison on synthetic data.*

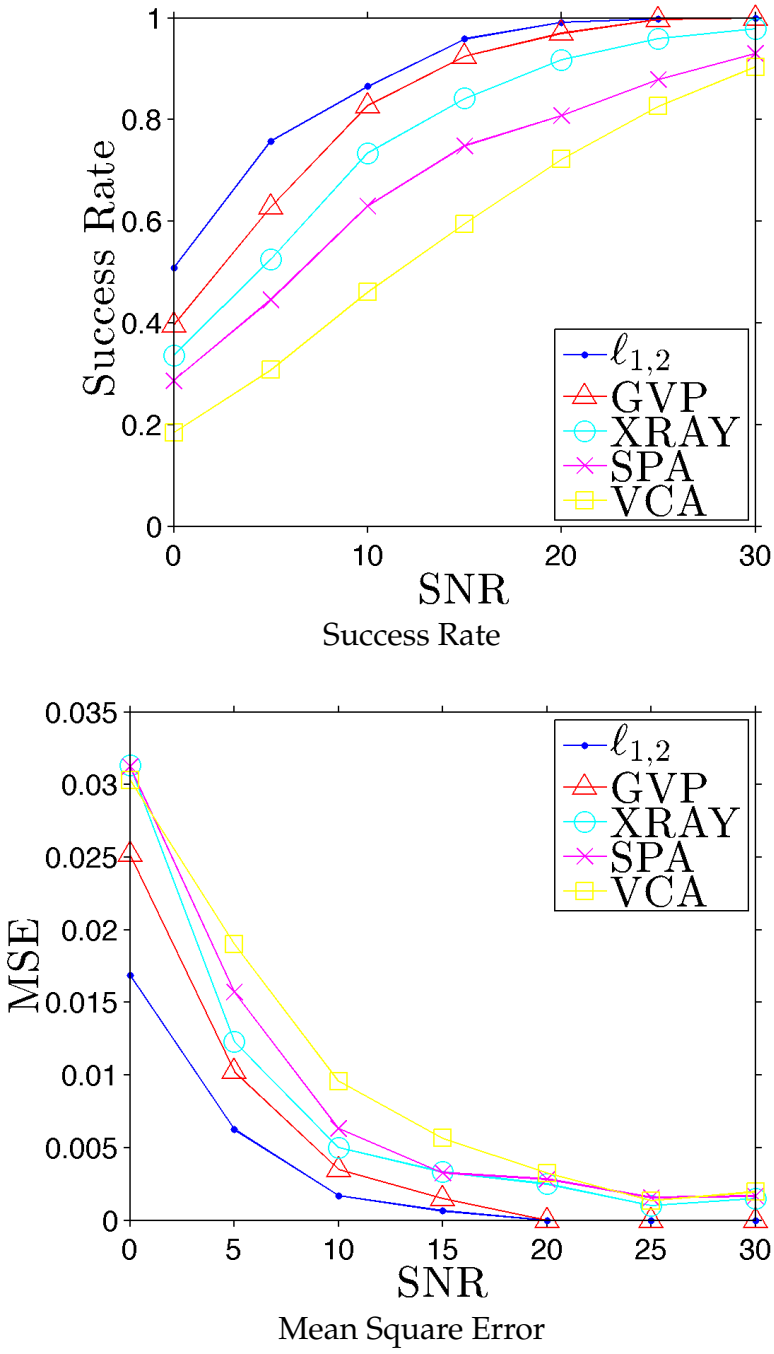
Algorithm	VCA	SPA	XRAY	GVP	$\ell_{1,2}$
Time	0.35	0.14	1.99	5.01	30.55

where each element of N is drawn from a Normal distribution. The signal–to–noise ratio (SNR), defined as $\text{SNR} = 10\log_{10}\left(\frac{1}{ns}\frac{\|\boldsymbol{Y}\|_F}{\|N\|_F}\right)$, is varied from 0 to 30 dB. For each SNR level, the simulation is repeated 100 times. The success rate and mean square error are shown in Figure 3.7. We can see that the GVP algorithm outperforms other greedy algorithms. Moreover, it is approximately 6 times faster than the $\ell_{1,2}$–minimization as shown in Table 3.2.

3.4.1.2 Hyperspectral unmixing

This subsection presents numerical results of the GVP algorithm when applied to the hyperspectral unmixing problem. We use the Urban data [2] in our experiments. This data is mainly constituted of six types of materials including road, roof, metal, dirt, grass, and tree. Additionally, the dimension of the preprocessed data cube is $307 \times 307 \times 162$ (Qu et al., 2014). The parameters of the original data matrix $\tilde{\boldsymbol{Y}} \in \mathbb{R}^{n\times m}$ are thus given by: the signal dimension $n = 162$, the total number of data points $m' = 307 \times 307 = 94249$, and the number of endmembers $s = 6$. We reduced the size of the dataset by merging similar columns, resulting in a reduced data matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times m}$ of size 162×1147. Algorithms are then applied to this reduced dataset to extract $s = 6$ endmembers. Fig 3.8 and Fig. 3.9 show the Urban image and its corresponding signatures, respectively. Figure 3.10 illustrates the vertices identified

[2] http://www.agc.army.mil/



Success Rate

Mean Square Error

Figure 3.7: *Robustness comparison on synthetic data.*

by various methods. Our GVP algorithm extracts distinct endmembers which are mostly similar to ones manually labeled.

Figure 3.8: *Urban image data.*

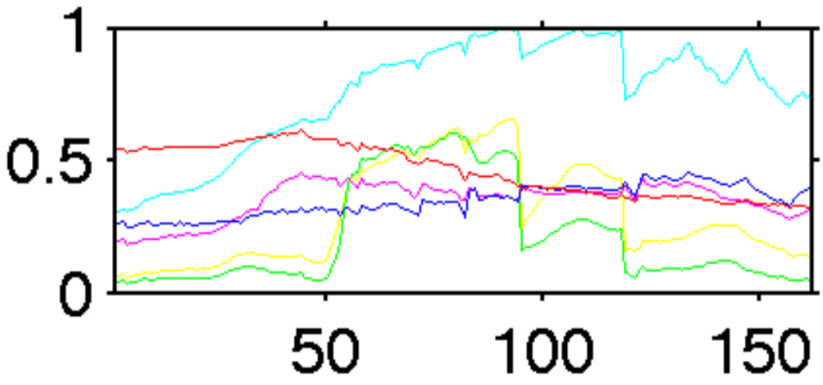


Figure 3.9: *Urban signature data.*

To further compare the performance of the algorithms, we use the root mean square error RMSE $= \frac{1}{\sqrt{ns}}\|\tilde{\boldsymbol{Y}} - \boldsymbol{Y}_{\mathcal{S}}\boldsymbol{X}\|_F$. Here, $\mathcal{S}$ is the endmember index set extracted from the reduced data matrix $\boldsymbol{Y}$ by the algorithms, and $\boldsymbol{X}$ is the

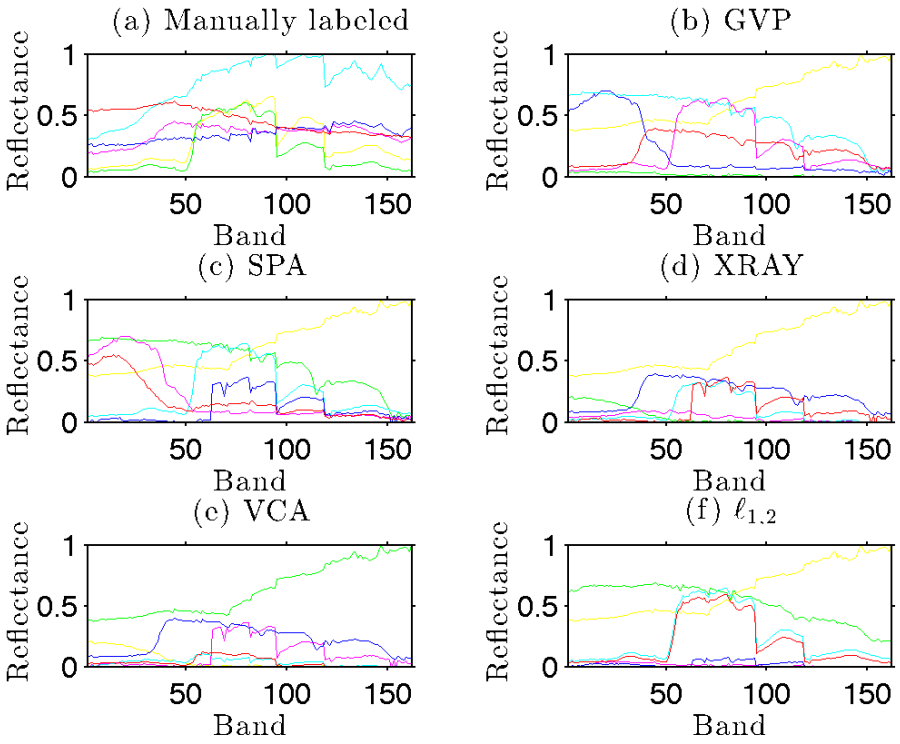


Figure 3.10: *Signatures obtained by manually labeling and by the algorithms.*

coefficient of the original data $\tilde{\boldsymbol{Y}}$ when projected onto the estimated polytope, i.e., $\boldsymbol{X} = \text{argmin} \|\tilde{\boldsymbol{Y}} - \boldsymbol{Y}_{\mathcal{S}}\boldsymbol{X}\|_F$ s.t. $\boldsymbol{X} \geq \boldsymbol{0}$ and $\boldsymbol{1}^T\boldsymbol{X} = \boldsymbol{1}^T$. This quantity measures the quality of the approximation: a small value of RMSE indicates that the detected polytope covers the entire data set well. Results for the Urban data set is shown in Table 3.3. It can be seen that the computationally-efficient GVP algorithm outperforms other greedy algorithms by a significant margin. In fact, its performance approaches that of $\ell_{1,2}$–minimization, while GVP is approximately 100 times faster than this convex method.

Table 3.3: *RMSE and running time comparison on Urban data.*

Algorithm	VCA	SPA	XRAY	GVP	$\ell_{1,2}$
RMSE	93.20	37.88	54.02	26.27	25.28
Time	0.11	0.098	0.24	2.23	199.71

3.4.2 Conclusion

We presents the GVP algorithm for choosing representative based on the convex hull assumption. GVP is fast, robust, and provably correct. We evaluate the proposed algorithm on both synthetic and real hyperspectral data, and show its superior performance compared with other state-of-the-art greedy pursuit algorithms.

3.5 Row entropy minimization

In this section, we address the intractability of (3.7) by proposing a non-convex relaxation to this problem. Specifically, we introduce a row sparsity measure based on the entropy function over the rows of the coefficient matrix. We show rigorously that by minimizing this measure under separability, one can robustly recover the vertices even when the data is corrupted by noise. As we will show in the experiment section, our algorithm is remarkably more robust than state-of-the-art algorithms for solving the representative selection problem under Assumption 1.

To begin, for any matrix $\boldsymbol{X} \in \mathbb{R}^{m \times m}$, define $\boldsymbol{\nu}(\boldsymbol{X}) = [\|\boldsymbol{x}^1\|_\infty, ..., \|\boldsymbol{x}^m\|_\infty]^\mathsf{T}$. Then the sparsity of $\boldsymbol{\nu}(\boldsymbol{X})$ and the row sparsity of $\boldsymbol{X}$ are equivalent. To overcome the NP-hardness of (3.7), we propose to solve the following optimization

problem named *Row Entropy Minimization (REM)*:

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{h,\infty} \quad \text{s.t.} \quad \boldsymbol{Y}\boldsymbol{X} = \boldsymbol{Y},\ \boldsymbol{X} \geq \boldsymbol{0},\ \boldsymbol{1}^\mathsf{T}\boldsymbol{X} = \boldsymbol{1}^\mathsf{T}, \tag{3.26}$$

where $\|\boldsymbol{X}\|_{\mathrm{h},\infty} = h(\boldsymbol{\nu}(\boldsymbol{X}))$. Here, the *entropy function* $h(\cdot)$ is defined as

$$h(\boldsymbol{z}) = -\sum_i \frac{|z_i|}{\|\boldsymbol{z}\|_1} \log \frac{|z_i|}{\|\boldsymbol{z}\|_1}, \tag{3.27}$$

for any vector $\boldsymbol{z} \in \mathbb{R}^m$. We adopt the convention that $0 \log 0 = 0$ and $h(0) = 0$. It was argued in (Tran et al., 2016; Huang, Tran, and Tran, 2016) that this function promotes the sparsity of its argument by skewing the signal energy towards a few of its elements. Therefore, a small value of the row entropy term $\|\boldsymbol{X}\|_{h,\infty}$ induces the row sparsity of $\boldsymbol{X}$.

In practice, data is often corrupted by noise. In this case, we consider the following noisy model

$$\tilde{\boldsymbol{Y}} = \boldsymbol{Y} + \boldsymbol{N} = \boldsymbol{Y}_{\mathcal{S}}\boldsymbol{Z} + \boldsymbol{N}, \tag{3.28}$$

where $\boldsymbol{Y}$, $\boldsymbol{Y}_{\mathcal{S}}$ and $\boldsymbol{Z}$ are defined in Assumption 1, and $\boldsymbol{N} \in \mathbb{R}^{n \times m}$ is a bounded noise matrix. Here, each column of the noise matrix is assumed to be bounded, i.e., $\|\boldsymbol{n}_j\|_2 \leq \epsilon$, for some small positive number ϵ, and for every column $\boldsymbol{n}_j$ of $\boldsymbol{N}$. We thus find the vertices in noisy settings by first solving the following

robust variant of REM:

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{X}\|_{h,\infty} \\ \text{s.t.} \quad & \|\tilde{\boldsymbol{y}}_j - \tilde{\boldsymbol{Y}}\boldsymbol{x}_j\|_2 \leq 2\epsilon, \forall j = 1, ..., m, \\ & \boldsymbol{X} \geq \boldsymbol{0},\ \boldsymbol{1}^\mathsf{T}\boldsymbol{X} = \boldsymbol{1}^\mathsf{T}. \end{aligned} \tag{3.29}$$

The vertices can then be identified from the dominant rows of the optimal solution of this optimization problem. This procedure is summarized in Algorithm 1. In the next section, we will show that under the convex hull

Algorithm 1 Robust REM for Vertex Identification

input: Noisy data matrix $\tilde{\boldsymbol{Y}}$, the noise level ϵ.
output: The estimated vertex set $\hat{\mathcal{S}}$ of the original data matrix.
1. Find the optimal solution $\boldsymbol{X}_*$ of the optimization problem (3.29).
2. Let $\hat{\mathcal{S}}$ be the index set corresponding to the s rows of $\boldsymbol{X}_*$ with the largest ℓ_∞ norm.

assumption, Algorithm 1 is guaranteed to exactly identify the vertices of the corrupted data matrix, when the noise power is relatively small. Before continuing, we would like to point out that the row entropy term $\|\cdot\|_{h,\infty}$ is not a norm as it does not satisfy the triangle inequality.

3.5.1 Theoretical guarantees

In this section, we prove that REM is robust under small perturbation. To simplify the analysis, we assume that the columns of $\boldsymbol{Y}$ are distinct. To begin,

we define the margin parameter

$$\rho = \min_{j \notin \mathcal{S}, k \in \mathcal{S}} \|\boldsymbol{y}_j - \boldsymbol{y}_k\|_2, \tag{3.30}$$

which characterizes the isolation of the vertices. We assume that $\rho > 0$, meaning the vertices are separated from the non-vertex data points. Furthermore, let

$$\gamma = \min_{k \in \mathcal{S}} \min_{\alpha \geq 0, \mathbf{1}^\mathsf{T}\alpha = 1} \|\boldsymbol{y}_k - \boldsymbol{Y}_{\mathcal{S}\setminus k}\alpha\|_2, \tag{3.31}$$

which bounds from below the distance from a vertex to the convex hull generated by the other vertices. In some sense, this parameter characterize the fatness of the polytope generated by the data vertices. Intuitively, large values of ρ and γ make the isolation of the vertices and the shape of the data polytope more robust to noise. This in turn makes it easy to identify the vertices. Fig 3.11 and Fig 3.12 illustrates this intuition. Finally, we assume that the data is bounded by a finite number defined by

$$\kappa = \max_{j=1,\ldots,m} \|y_j\|_2. \tag{3.32}$$

It's shown in (Tran et al., 2015) that this maximum value is attained at one of the vertices. Therefore, it can be rewritten as $\kappa = \max_{j \in \mathcal{S}} \|y_j\|_2$.

We are now ready to state our main result.

Theorem 14. *Let $\boldsymbol{Y}$ be a data matrix satisfying Assumption 1. Suppose the data is concatenated by bounded noise, i.e., $\tilde{\boldsymbol{Y}} = \boldsymbol{Y} + \boldsymbol{N}$, where $\|\boldsymbol{n}_j\|_2 \leq \epsilon, \forall j = 1, \ldots, m$. If*

$$\epsilon < \frac{\rho\gamma}{8\kappa(s+1)}, \tag{3.33}$$

then Algorithm 1 identifies the vertices of $\boldsymbol{Y}$ exactly.

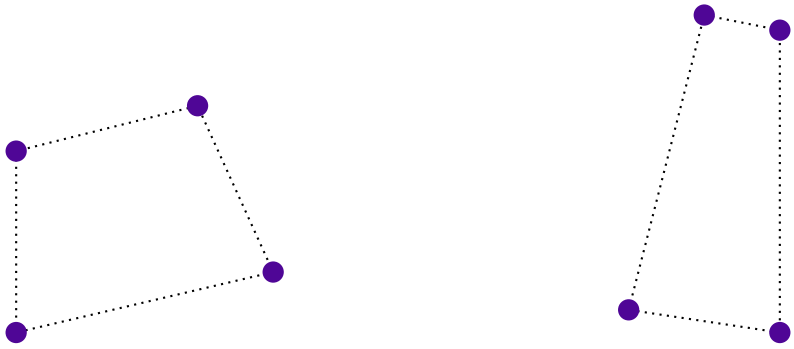


Figure 3.11: *The fatness parameter γ dictates the fatness of the data polytope.* **Left:** A fat data polytope (large γ). **Right:** A thin data polytope (small γ).

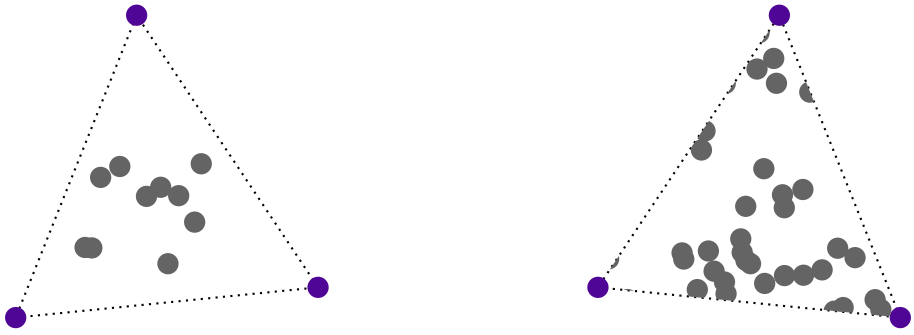


Figure 3.12: *The margin parameter κ characterizes the isolation of the vertices relatively to the data energy.* **Left:** A polytope with strongly isolated vertices, i.e., large ρ/κ. **Right:** A polytope with weakly isolated vertices, i.e., small ρ/κ.

The proof for the main theorem is given at the end of this section. The main ingredient of our analysis is the *concentration* property of the entropy function. The following lemmas formalize this important property.

Lemma 15. *Let* $\boldsymbol{x} \in \mathbb{R}_+^m$ *such that* $0 \leq x_i \leq 1, \forall 1 \leq i \leq m$. *Let* k *and* l *be two arbitrary distinct indices satisfying* $\boldsymbol{x}_k \geq x_l$. *Define* $\boldsymbol{x}(\delta) := \tilde{\boldsymbol{x}}$ *as*

$$\tilde{x}_i = x_i, \ \forall i \neq k, l; \quad \tilde{x}_k = \boldsymbol{x}_k + \delta, \quad \tilde{x}_l = x_l - \delta,$$

where δ *is a small positive number such that* $0 \leq \tilde{x}_k, \tilde{x}_l \leq 1$. *Then* $h(\boldsymbol{x}) > h(\boldsymbol{x}(\delta))$.

In words, concentrating signal energy on significant elements while dispersing energy from small elements decreases the value of the entropy function. Fig 3.13 visualizes this property of the entropy function.

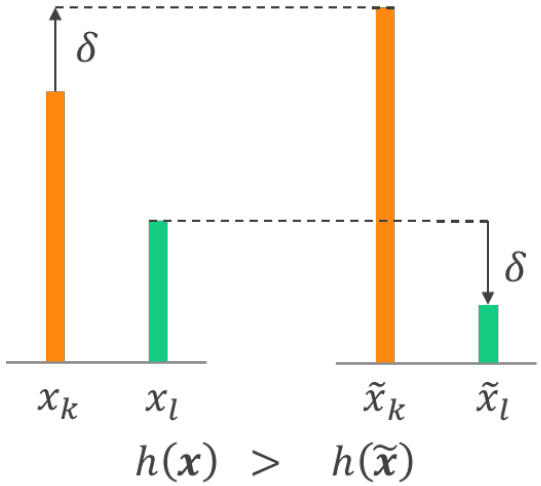


Figure 3.13: *Concentration property of the entropy function*. Concentrating signal energy on significant elements while dispersing energy from small elements decreases the value of the entropy function.

Proof. First of all, $h(\boldsymbol{x}(0)) = h(\boldsymbol{x})$. Therefore, by the continuity of the entropy function, it suffices to show that $h(\boldsymbol{x}(\delta))$ is a strictly monotonically decreasing function of δ. To do so, notice that $\|\tilde{\boldsymbol{x}}\|_1 = \|\boldsymbol{x}\|_1$ and

$$\begin{aligned} h(\boldsymbol{x}(\delta)) &= -\sum_i \frac{\tilde{x}_i}{\|\tilde{\boldsymbol{x}}\|_1} \log \frac{\tilde{x}_i}{\|\tilde{\boldsymbol{x}}\|_1} \\ &= -\sum_{i \neq k,l} \frac{x_i}{\|\boldsymbol{x}\|_1} \log \frac{x_i}{\|\boldsymbol{x}\|_1} - \frac{\boldsymbol{x}_k + \delta}{\|\boldsymbol{x}\|_1} \log \frac{\boldsymbol{x}_k + \delta}{\|\boldsymbol{x}\|_1} - \frac{x_l - \delta}{\|\boldsymbol{x}\|_1} \log \frac{x_l - \delta}{\|\boldsymbol{x}\|_1}. \end{aligned}$$

Define $g(\gamma) \coloneqq h(\boldsymbol{x}(\delta))$, then

$$\begin{aligned} g'(\delta) &= -\frac{1}{\|\boldsymbol{x}\|_1} \log \frac{\boldsymbol{x}_k + \delta}{\|\boldsymbol{x}\|_1} - \frac{\boldsymbol{x}_k + \delta}{\|\boldsymbol{x}\|_1} \frac{1}{\|\boldsymbol{x}\|_1} \frac{\|\boldsymbol{x}\|_1}{\boldsymbol{x}_k + \delta} \\ &\quad + \frac{1}{\|\boldsymbol{x}\|_1} \log \frac{x_l - \delta}{\|\boldsymbol{x}\|_1} + \frac{x_l - \delta}{\|\boldsymbol{x}\|_1} \frac{1}{\|\boldsymbol{x}\|_1} \frac{\|\boldsymbol{x}\|_1}{x_l - \delta} \\ &= \frac{1}{\|\boldsymbol{x}\|_1} \log \frac{x_l - \delta}{\boldsymbol{x}_k + \delta}. \end{aligned} \tag{3.34}$$

As $x_l \leq \boldsymbol{x}_k$ and $\delta > 0$, it follows that $g'(\delta) < 0$, and thus $g(\delta)$ is strictly monotonically decreasing. We conclude that $h(\boldsymbol{x}) = h(\boldsymbol{x}(0)) > h(\boldsymbol{x}(\delta))$, for any small positive δ. □

Lemma 16. *Let* $\boldsymbol{x} = \begin{pmatrix} \mathbf{1}_k^\mathsf{T} & \mathbf{0}_{m-k}^\mathsf{T} \end{pmatrix}^\mathsf{T}$, *for some* $1 \leq k \leq m-1$, *and* $\boldsymbol{x}(\alpha) = \begin{pmatrix} \mathbf{1}_k^\mathsf{T} & \alpha & \mathbf{0}_{m-k-1}^\mathsf{T} \end{pmatrix}^\mathsf{T}$, *for some* $0 < \alpha \leq 1$. *It follows that* $h(\boldsymbol{x}) < h(\boldsymbol{x}(\alpha))$.

Proof. To begin, denote $g(\alpha) \coloneqq h(\boldsymbol{x}(\alpha))$, then

$$g(\alpha) = \log(k + \alpha) - \frac{\alpha}{k + \alpha} \log \alpha$$

If $\alpha = 1$,

$$g(\alpha) = \log(k+1) > \log k = h(\boldsymbol{x}).$$

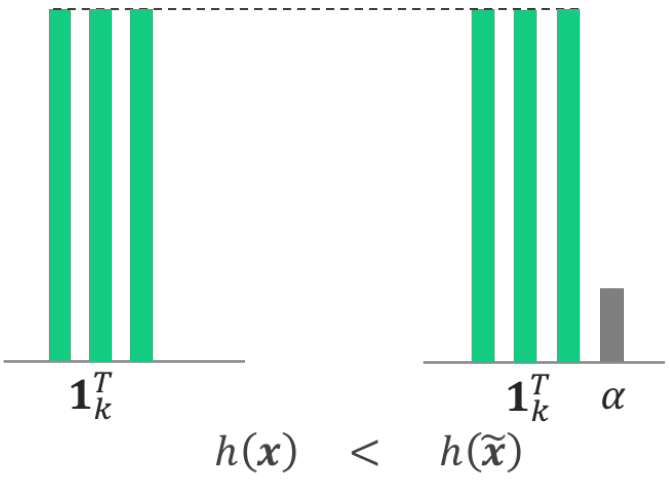


Figure 3.14: *Sparse promoting property of the entropy function.*

On the other hand, if $0 < \alpha < 1$, then

$$g'(\alpha) = -\frac{k}{(k+\alpha)^2}\log\alpha > 0, \forall 0 < \alpha < 1.$$

This implies $g(\alpha)$ is strictly monotonically increasing. By the continuity of the entropy function, it follows that $h(\boldsymbol{x}) = h(\boldsymbol{x}(0)) < h(\boldsymbol{x}(\alpha))$, for any $0 < \alpha \leq 1$. □

Intuitively, the aforementioned lemmas suggest that when the vector elements are bounded from above, solutions of entropy function minimization tend to concentrate the energy on the least number of elements. This is formalized in the lemma below. Its proof can be obtained by iteratively applying Lemma 15 and Lemma 16.

Lemma 17. *Let* $\boldsymbol{x} = \begin{pmatrix}\mathbf{1}_k^\mathsf{T} & \mathbf{0}_{m-k}^\mathsf{T}\end{pmatrix}^\mathsf{T}$, *and* $\tilde{\boldsymbol{x}} = \begin{pmatrix}\mathbf{1}_k^\mathsf{T} & \boldsymbol{\alpha}^\mathsf{T}\end{pmatrix}^\mathsf{T}$, *where* $\boldsymbol{\alpha} \in \mathbb{R}_+^{m-k}$. *If* $\boldsymbol{\alpha}$ *is nonzero, then* $h(\boldsymbol{x}) < h(\tilde{\boldsymbol{x}})$.

It can be easily seen from these lemmas that that the row entropy norm promotes row sparsity. In particular, spreading the row energy of a matrix leads to its high row entropy norm whereas concentrating its row energy decreases its row entropy norm $\|\cdot\|_{h,\infty}$. We call this property the *row Schur concavity* property of the row entropy norm $\|\cdot\|_{h,\infty}$ which resembles the Schur concavity of vectors. This property is illustrated in Fig 3.15.

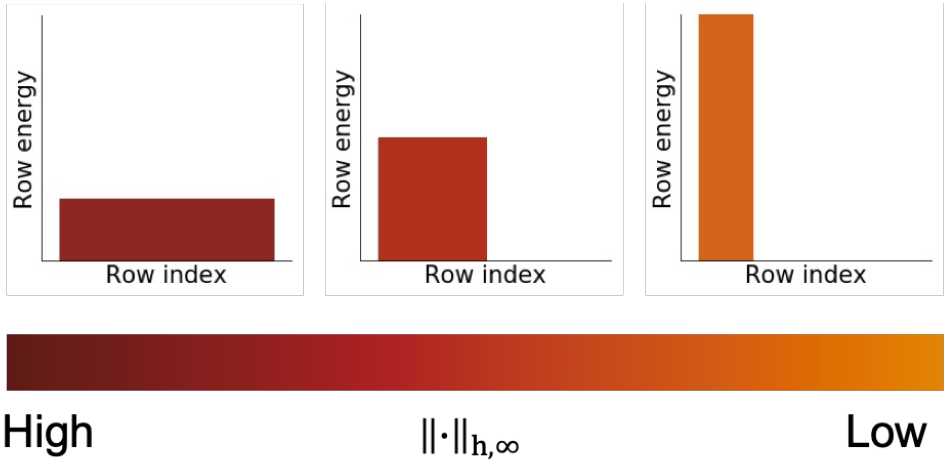


Figure 3.15: *Row Schur concavity property of row entropy norm* $\|\cdot\|_{h,\infty}$. Spreading the row energy of a matrix leads to its high row entropy norm whereas concentrating its row energy decreases its row entropy norm.

We now prove Theorem 14.

Proof. Consider the noisy model (3.28), where $\boldsymbol{Y}$ satisfies Assumption 1, and $\boldsymbol{N}$ is a bounded noise matrix whose column energy is bounded by $\epsilon > 0$. Let $\boldsymbol{X}$ be a feasible solution of (3.29). Similar to the proof of Theorem 1 in (Xiao Fu, 2016), we can show that

$$\|\boldsymbol{x}^k\|_\infty \geq x_{kk} \geq 1 - \frac{8\epsilon\kappa}{\rho\gamma}, \forall k \in \mathcal{S}. \tag{3.35}$$

Let $\boldsymbol{Z}$ be the coefficient matrix defined in Assumption 1, then $\bar{\boldsymbol{Z}} = \begin{bmatrix}\boldsymbol{Z}^\mathsf{T} & \boldsymbol{0}\end{bmatrix}^\mathsf{T}$

is a feasible solution of (3.29). Let $\boldsymbol{X}_*$ be the optimal solution of (3.29). It follows that $\|\boldsymbol{X}_*\|_{h,\infty} \leq \|\bar{\boldsymbol{Z}}\|_{h,\infty}$. Iteratively applying Lemmas 15, 16, and 17, this implies, for all $j \notin \mathcal{S}$,

$$\|\boldsymbol{x}_*^j\|_\infty \leq \sum_{j \notin \mathcal{S}} \|\boldsymbol{x}_*^j\|_\infty \leq s - \sum_{k \in \mathcal{S}} \|\boldsymbol{x}_*^k\|_\infty \leq \frac{8\epsilon\kappa}{\rho\gamma} s. \tag{3.36}$$

Therefore, if $\frac{8\epsilon\kappa}{\rho\gamma} s < 1 - \frac{8\epsilon\kappa}{\rho\gamma}$, or equivalently, $\epsilon < \frac{\rho\gamma}{8\kappa(s+1)}$, then $\|\boldsymbol{x}_*^j\|_\infty < \|\boldsymbol{x}_*^k\|_\infty, \forall j \notin \mathcal{S}, k \in \mathcal{S}$. In other words, the s rows of $\boldsymbol{X}_*$ with the largest ℓ_∞ norm correspond to the vertices of the dataset. This completes the proof for the theorem. □

3.5.2 Iterative Algorithms for REM

As we will show in the experiment section, solving robust REM leads to better solutions comparing to state-of-the-art algorithms for choosing representatives under the convex hull assumption. The trade-off of is that the row entropy objective $\|\cdot\|_{h,\infty}$ in REM is nonconvex due to the non-convexity of the entropy function. We thus approximate the objective function by its first order approximation and utilize an iterative algorithm to solve a series of easier subproblems.

In a simplified setting, we denote $\boldsymbol{\nu} = \boldsymbol{\nu}(\boldsymbol{X})$ and $\boldsymbol{\nu}^t = \boldsymbol{\nu}(\boldsymbol{X}^t)$. Let $\boldsymbol{X}^t$ be the solution estimate at iteration t of the algorithm, then the first order

approximation of the objective function in REM is given by

$$\|\boldsymbol{X}\|_{h,\infty} = h(\boldsymbol{\nu}) \approx h(\boldsymbol{\nu}^t) + \nabla h(\boldsymbol{\nu}^t)^\mathsf{T}(\boldsymbol{\nu} - \boldsymbol{\nu}^t) \tag{3.37}$$

$$= \sum_i [\nabla h(\boldsymbol{\nu}^t)]_i \nu_i + h(\boldsymbol{\nu}^t) - \nabla h(\boldsymbol{\nu}^t)^\mathsf{T} \boldsymbol{\nu}^t.$$

Recall that $\nu_i = \|\boldsymbol{x}^i\|_\infty$, $\forall 1 \leq i \leq m$, the next solution estimate can thus be obtained by solving

$$\min_{\boldsymbol{X}} \sum_i w_i^t \|\boldsymbol{x}^i\|_\infty \quad \text{s.t.} \quad \boldsymbol{Y}\boldsymbol{X} = \boldsymbol{Y},\ \boldsymbol{X} \geq \boldsymbol{0},\ \boldsymbol{1}^\mathsf{T}\boldsymbol{X} = \boldsymbol{1}^\mathsf{T}, \tag{3.38}$$

where $w_i^t = [\nabla h(\boldsymbol{\nu}^t)]_i$, $\forall 1 \leq i \leq m$. The following proposition shows that the weights has a closed and easy-to-compute form (Tran et al., 2016).

Proposition 18. *Let h be the entropy function defined in (3.27), and let $\boldsymbol{\nu}$ be a nonzero nonnegative vector, then*

$$\frac{\partial h(\boldsymbol{\nu})}{\partial \nu_i} = -\frac{\log \nu_i}{\|\boldsymbol{\nu}\|_1} + \frac{\sum_j \nu_j \log \nu_j}{\|\boldsymbol{\nu}\|_1^2}. \tag{3.39}$$

As a consequence,

$$w_i^t = -\frac{\log \nu_i^t}{\|\boldsymbol{\nu}^t\|_1} + \frac{\sum_j \nu_j^t \log \nu_j^t}{\|\boldsymbol{\nu}^t\|_1^2}, \tag{3.40}$$

for $\nu_i^t > 0$. When $\nu_i^t = 0$, we let $w_i^t = +\infty$. Moreover, the weights are dictated by the concentration behavior of the entropy function minimization. The following corollary summarizes this insight. It follows by the fact that $0 \leq \nu_i^t \leq 1$, $\forall 1 \leq i \leq m$.

Corollary 19. *If $\nu_i^t < \nu_{k'}^t$, then $w_i^t > w_k^t$.*

In other words, small energy rows are given large weights at the next iteration,

and are thus further suppressed. Therefore, at the end of the algorithm energy is concentrated only on a small subset of rows.

In noisy settings, the subproblems in robust REM can be written as

$$\min_{\boldsymbol{X}} \quad \lambda \sum_i w_i^t \|\boldsymbol{x}^i\|_\infty \tag{3.41}$$

$$\text{s.t.} \quad \|\tilde{\boldsymbol{y}}_j - \tilde{\boldsymbol{Y}} \tag{3.42}$$

$$varColj\|_2 \leq 2\epsilon, \forall j = 1, ..., m,$$

$$\boldsymbol{X} \geq \boldsymbol{0},\ \boldsymbol{1}^\mathsf{T}\boldsymbol{X} = \boldsymbol{1}^\mathsf{T}.$$

Therefore, at each iteration of REM and its robust variant, we solve a weighted $\ell_{1,\infty}$ subproblem under the same constraints as the original problem. Problems (3.38) and (3.41) can be solved efficiently by an Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). The main steps of this iterative algorithm are summarized in Algorithm 2.

Algorithm 2 Iterative Algorithms for Solving Robust REM

input: data matrix $\boldsymbol{Y}$, the noise level ϵ.
initialization: $\boldsymbol{X}^0$.
while not converged **do**
1. *Update the weights*:

$$w_i^t = -\frac{\log v_i^t}{\|\boldsymbol{v}^t\|_1} + \frac{\sum_j v_j^t \log v_j^t}{\|\boldsymbol{v}^t\|_1^2},\ i = 1, ..., m. \tag{3.43}$$

2. *Update the estimate*: Set $\boldsymbol{X}^{t+1}$ to be the optimal solution of (3.41).
end while
output: Estimated solution $\boldsymbol{X}_* = \boldsymbol{X}^t$.

3.5.3 Experimental Results

This section presents experimental results for REM algorithm on both synthetic and real data. In the synthetic data experiment, we test the robustness of our algorithm in finding the vertices of a dataset, and benchmark it against various state-of-the-art algorithms. For the benchmarked algorithms, we use the implementations on the author websites. In the real data experiments, we apply REM to the video and text summarization problem.

3.5.3.1 Vertex recovery on synthetic data

We test the robustness of our proposed algorithm against noise on a synthetic dataset. The experiment setting is similar to that in (Gillis, 2014). For each simulation, the data is generated as follows. Elements of each column of the vertex matrix $\boldsymbol{Y}_{\mathcal{S}} \in \mathbb{R}^{n \times s}$ are sampled from a uniform distribution on $[0, 1]$. The coefficient matrix $\boldsymbol{Z} \in \mathbb{R}^{s \times m}$ has the form of $[\boldsymbol{I}_s, \boldsymbol{Z}']$ where $\boldsymbol{I}_s \in \mathbb{R}^{s \times s}$ is the

identity matrix, and each column of $\boldsymbol{Z}' \in \mathbb{R}_+^{m \times (m-s)}$ follows from a Dirichlet distribution whose parameters are chosen from a uniform distribution on $[0, 1]$. The data matrix is generated by $\boldsymbol{Y} = \boldsymbol{Y}_S \boldsymbol{Z} + \boldsymbol{N}$ where each element of the noise matrix $\boldsymbol{N}$ is drawn from a Normal distribution, then is multiplied by some parameter β. Throughout the experiments, we let $n = 5, m = 25$, and $s = 5$.

Figure 3.16 shows the ℓ_∞ norm of the rows of typical solutions of REM when the data is corrupted by moderate and large noise. It can be seen that the energy of the solutions concentrates on the rows corresponding to the vertices, which is consistent with Theorem 14.

We next compare our proposed algorithm with state-of-the-art representative selection algorithms: XRAY (Kumar, Sindhwani, and Kambadur, 2012), SPA (Gillis and Vavasis, 2014), SNPA (Gillis, 2014), and GVP (Tran et al., 2015). Figure 3.17 shows the exact recovery rates of the algorithms. It can be seen that REM is significantly more robust than the others.

3.5.3.2 Video summarization

We demonstrate in this subsection the efficacy of REM in summarizing videos. In this problem, given a video sequence, our objective is to identify a few key frames of the video allowing one to infer the main content of the video.

We consider a soccer video obtained from the internet. A subset of frames are shown in Fig. 3.18. The video consists of multiple shots of different scenes. Each shot itself consists of a series of activities. In Fig. 3.18, we show 2 frames before and 2 frames after each representative frame in a purple box, chosen by

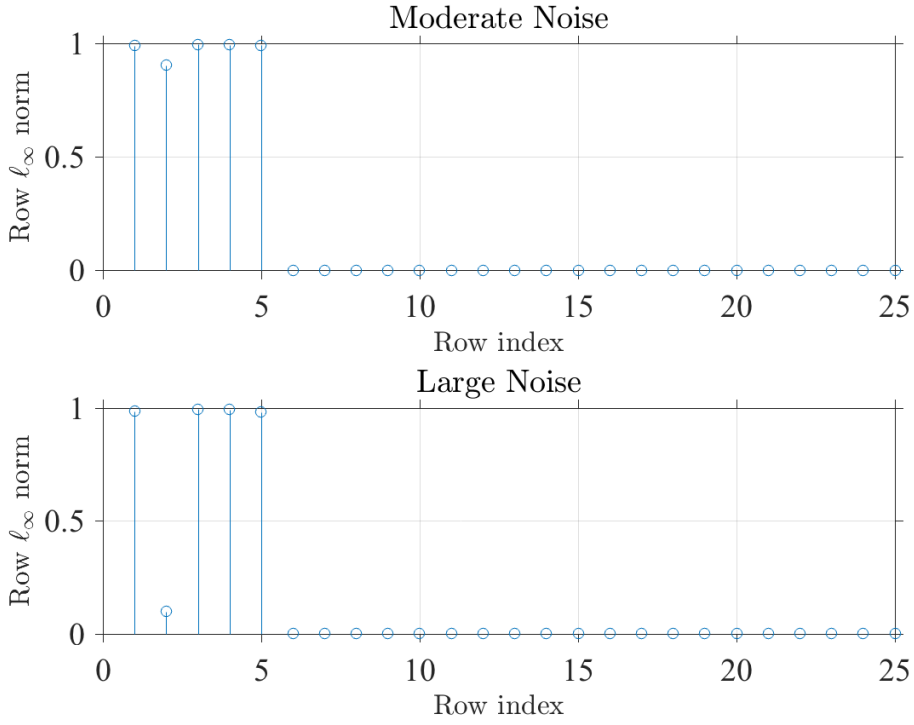


Figure 3.16: *Row ℓ_∞ norm of typical solutions of REM*. **Top**: moderate noise. **Bottom**: large noise.

our algorithm REM, along with representative frame itself. We transform the video tensor into a 2D matrix each column of which is the vectorized version of a scene. We then apply our REM algorithm to this 2D matrix and obtain 7 representatives, which are 2D frames in purple boxes in Fig. 3.18. It can be seen that these representatives summarize nicely key events of this soccer video segment which is the highlight of a soccer goal. In particular, the first exemplar shows the player is about to take a shot. The second one indicates that he scored a goal. The third and forth representatives show he is running and celebrating the goal. In the next two exemplars, his teammates join his celebration. Finally, in the final chosen scene, the audience is cheering the

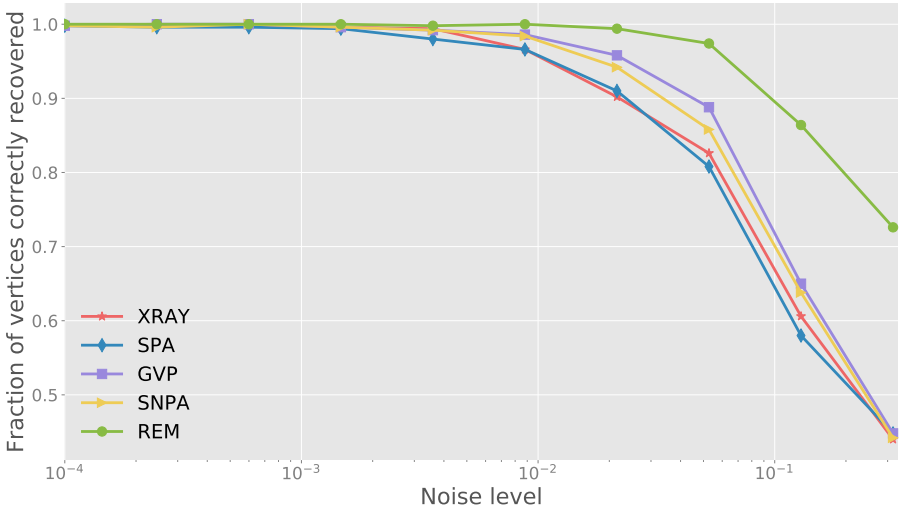


Figure 3.17: *Robustness comparison on synthetic data.*

goal.

We compared our summarization result against that produced by the $\ell_{1,q}$ minimization approach (Elhamifar, Sapiro, and Vidal, 2012) as illustrated in Fig 3.19 and Fig 3.20. The figures show that the representatives chosen by REM are significantly different each of which shows a scene change in the video whereas there are several similar representatives in the summarization obtained by $\ell_{1,q}$ minimization. This implies that REM produces more effective summarization than that produced by the convex relaxation approach.

3.5.3.3 Amazon review summarization

In this subsection, we apply REM to the problem of summarizing Amazon reviews. Given a collection of reviews of an Amazon product, we aim to choose a small subset of representative reviews that well describe this item.

Figure 3.18: *Video summarization result produced by REM.*

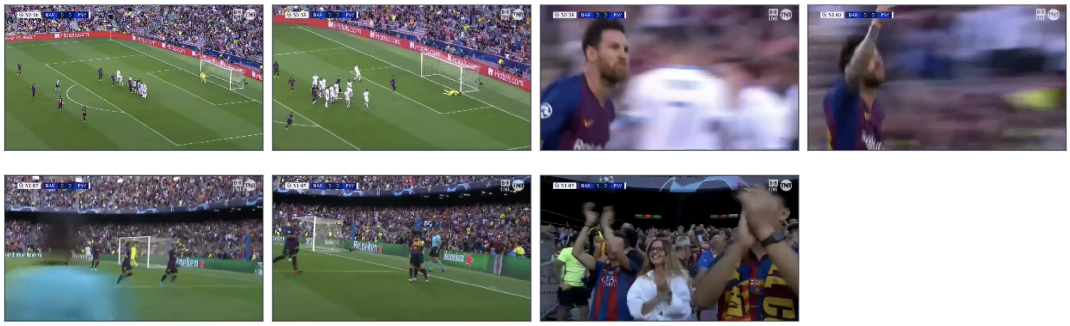


Figure 3.19: *Representative frames produced by REM*. The representatives are significantly different each of which shows a scene change in the video.

Amazon review dataset. We use the Amazon review dataset curated by

Figure 3.20: *Representative frames chosen by $\ell_{1,q}$ minimization (Elhamifar, Sapiro, and Vidal, 2012).* There are several similar representatives in the summarization.

McAuley et. al. [3]. The dataset contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014. It includes reviews (ratings, text, helpfulness votes), product metadata (descriptions, category information, price, brand, and image features), and links (also viewed/also bought graphs). As our purpose is to produce summarization for products that have a large number of reviews, we use the $5-$core subsets in which all users and items have at least 5 reviews and consider the most reviewed products.

The most reviewed products have several thousands of reviews. This makes decision making based on reading all of the reviews time consuming. Fig. 3.21 shows 10 first reviews and ratings of the most reviewed electronics product in this dataset.

We apply our REM algorithm to choose a few typical reviews of a certain product. The chosen product is an electronic device which has 308 reviews

[3] http://jmcauley.ucsd.edu/data/amazon/

	Review Text	Rating
1336614	No issues.	4
1336615	Purchased this for my device, it worked as advertised. You can never have too much phone memory, since I download a lot of stuff this was a no brainer for me.	5
1336616	it works as expected. I should have sprung for the higher capacity. I think its made a bit cheesier than the earlier versions; the paint looks not as clean as before	4
1336617	This think has worked out great.Had a diff. bran 64gb card and if went south after 3 months.This one has held up pretty well since I had my S3, now on my Note3.*** update 3/21/14I've had this for a few months and have had ZERO issue's since it was transferred from my S3 to my Note3 and into a note2. This card is reliable and solid!Cheers!	5
1336618	Bought it with Retail Packaging, arrived legit, in a orange envelope, english version not asian like the picture shows. arrived quickly, bought a 32 and 16 both retail packaging for my htc one sv and Lg Optimus, both cards in working order, probably best price you'll get for a nice sd card	5
1336619	It's mini storage. It doesn't do anything else and it's not supposed to. I purchased it to add additional storage to my Microsoft Surface Pro tablet which only come in 64 and 128 GB. It does what it's supposed to and SanDisk has a long standing reputation that speaks for itself.	5
1336620	I have it in my phone and it never skips a beat. File transfers are speedy and have not had any corruption issues or memory fade issues as I would expect from the Sandisk brand. Great card to own. Why entrust your precious files to a slightly cheaper piece of crap? If you lose everything can you forgive yourself for not spending the extra couple bucks on a trusted product that goes through good QA?	5
1336621	It's hard to believe how affordable digital has become. 32 GB in a device one quarter the sie of postage stamp would have been science fiction less than a generation ago.I picked this up for portable music when I didn't want to schlep (or risk) a phone or iPod. Works great with all SD card readers.Select with confidence.	5
1336622	Works in a HTC Rezound. Was running short of space on a 64GB Sandisk so I ordered this when it came out, fast and no issues.	5
1336623	in my galaxy s4, super fast card, and am totally happy, not happy having to still type to fill the required words though	5

Figure 3.21: *Ten first reviews and ratings of the most reviewed electronics product in the Amazon review dataset.*

or ratings. Fig. 3.22 shows the histogram of the ratings of this product. The histogram shows an overwhelming number of positive reviews of the product. This poses a great challenge for summarization algorithms. In particular, it is difficult to output a negative review due to the insignificant number of

negative reviews. It is likely that a summarization algorithm ignores negative reviews just by chance.

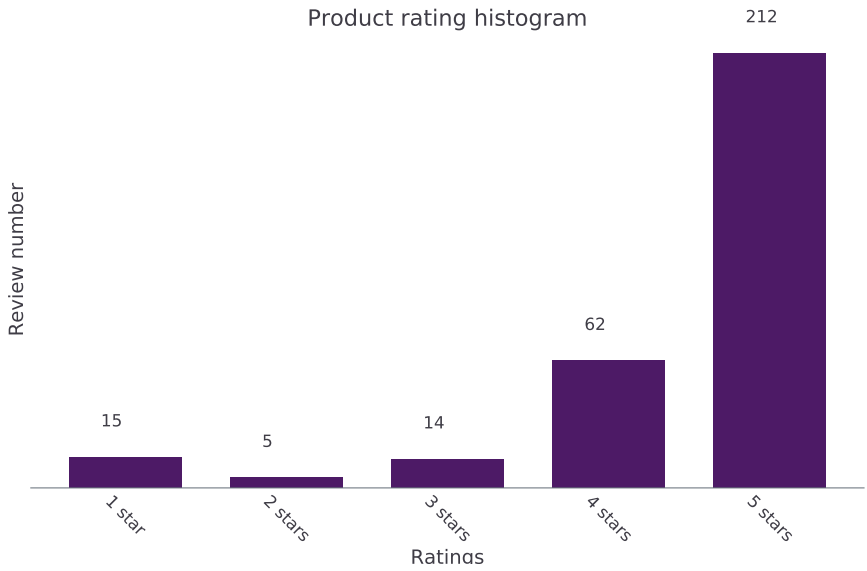


Figure 3.22: *Rating histogram of the summarized product.*

Fig. 3.23 shows the word-cloud representation of the reviews of this product. It highlights the key words in the review collection. The cloud give greater prominence to words that appear more frequently in the reviews. This representation gives good insight into the data, which resembles the word-counting idea behind popular feature selection technique such as Principal Component Analysis (PCA) and Dictionary Learning (DL). For example, one can infer from the word-cloud that this product is an optical drive with a certain number of possibly positive properties. However, it lacks of physical context. For instance, the word "burn" in the word-cloud is confusing. It seems to imply that this optical drive can be used to burn disc. As we will see in the actual reviews, this is not always the case.

Figure 3.23: *Wordcloud representation of the summarized product.*

Fig. 3.26 shows the representatives obtained by our REM algorithm applied to the considered product. First of all, the exemplars are consistent with several aspect of both the rating histogram and the word-cloud of the reviews. In particular, the positive representatives are overwhelming. It also features a very negative review. Each exemplar indicates a few properties of the product which in some sense are consistent with the word-cloud. In particular, the first review shows that this product is fast and easy to install. The second review indicate that the product is reliable, quite, cheap, and burn cds and dvds fast. The third one implies this is a SATA drive. The next exemplar shows that this is a good optical drive, and so on. The words featured in these representatives appear rather often in the reviews as shown in the word-cloud, which signal that the representatives are indeed reliable.

An advantage of the representatives over word-counting based techniques,

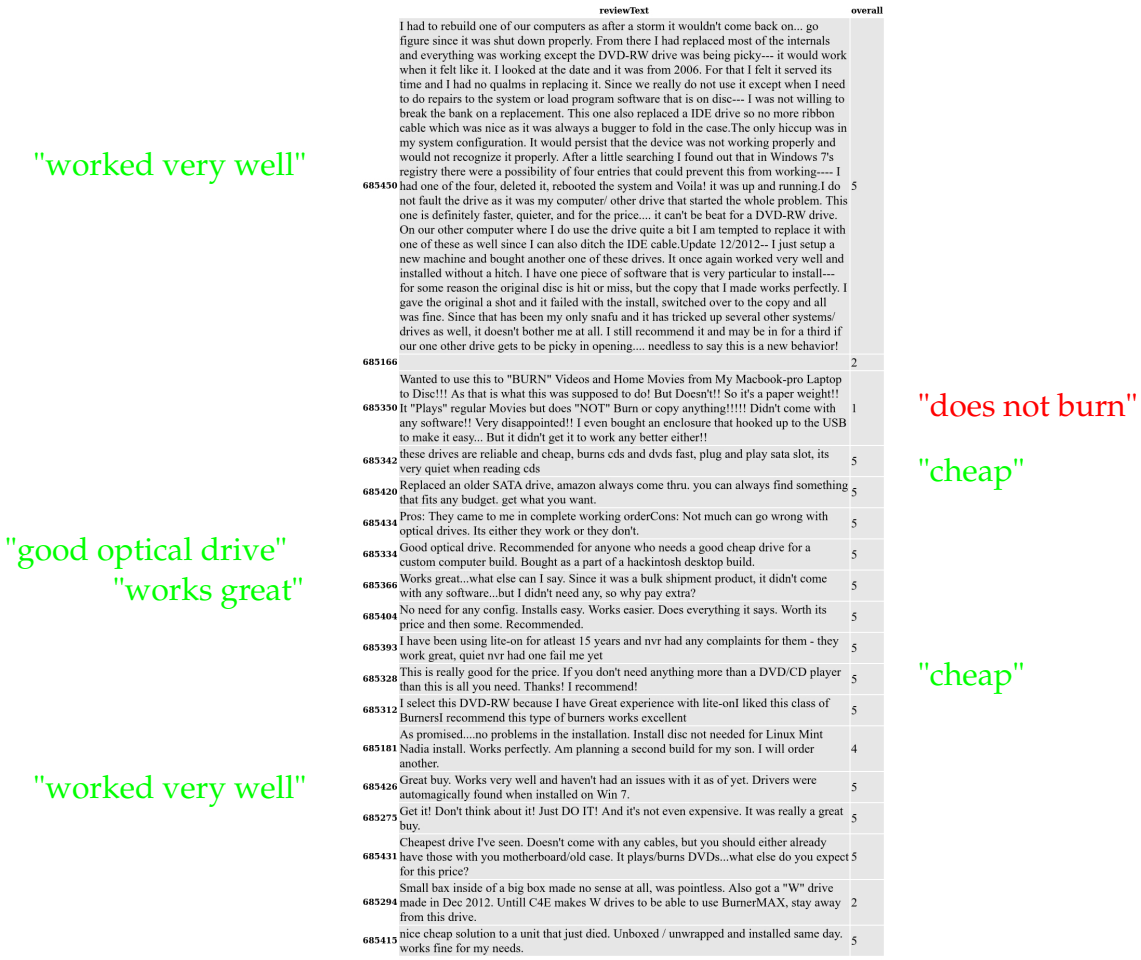
such as Principal Component Analysis (PCA) and Dictionary Learning (DL), is that they offer physical contexts. This is demonstrated vividly in the sixth exemplar with 1-star rating. More specifically, it shows that the burn feature of the product sometimes fails to work. This important characteristic of the product is missing in the word-cloud representation.

We compare the representatives selected by REM against ones obtained by the convex relaxation approach (Elhamifar, Sapiro, and Vidal, 2012). Fig 3.24 shows the representatives obtained by solving an $\ell_{1,q}$ minimization problem. It can be seen that some properties of the product appear in multiple different reviews. This implies that this convex relaxation approach produces a less effective summarization compared with that given by REM in which the representatives describe distinct properties of the product. . Furthermore, when we regularize the loss function of the $\ell_{1,q}$ minimization program to reduces the number of representatives, the algorithm ignores the negative reviews. This can be seen in Fig 3.25. This is due to the fact that there is an overwhelming number of positive reviews as shown in Fir 3.22.

In contrast, despite choosing a smaller number of representatives, REM is able to choose both negative and not-so-positive reviews which offer important information regarding the product. This implies that the summarization given by REM is more representative than that obtained by $\ell_{1,q}$ minimization.

3.5.4 Conclusion

In this section, we propose a row sparse model, namely Row Entropy Minimization, based on the entropy function to pick data vertices as representatives.



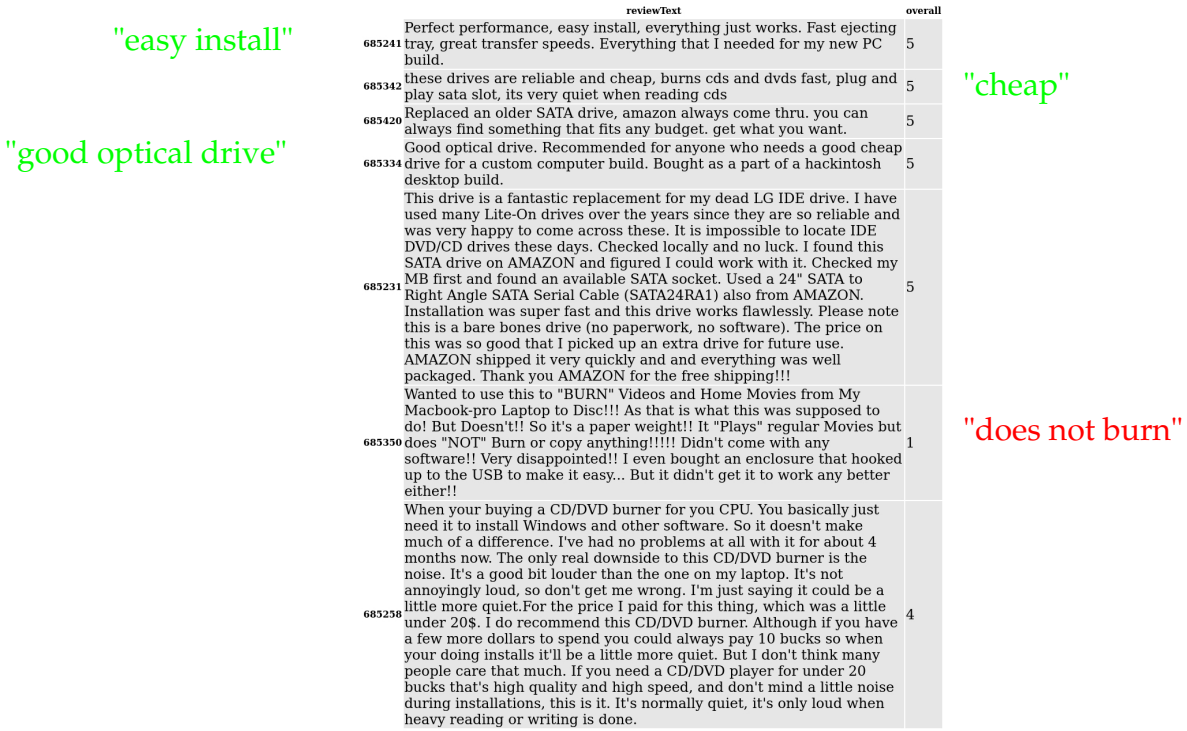
	reviewText	overall
685450	I had to rebuild one of our computers as after a storm it wouldn't come back on... go figure since it was shut down properly. From there I had replaced most of the internals and everything was working except the DVD-RW drive was being picky--- it would work when it felt like it. I looked at the date and it was from 2006. For that I felt it served its time and I had no qualms in replacing it. Since we really do not use it except when I need to do repairs to the system or load program software that is on disc--- I was not willing to break the bank on a replacement. This one also replaced a IDE drive so no more ribbon cable which was nice as it was always a bugger to fold in the case.The only hiccup was in my system configuration. It would persist that the device was not working properly and would not recognize it properly. After a little searching I found out that in Windows 7's registry there were a possibility of four entries that could prevent this from working---- I had one of the four, deleted it, rebooted the system and Voila! it was up and running.I do not fault the drive as it was my computer/ other drive that started the whole problem. This one is definitely faster, quieter, and for the price.... it can't be beat for a DVD-RW drive. On our other computer where I do use the drive quite a bit I am tempted to replace it with one of these as well since I can also ditch the IDE cable.Update 12/2012-- I just setup a new machine and bought another one of these drives. It once again worked very well and installed without a hitch. I have one piece of software that is very particular to install--- for some reason the original disc is hit or miss, but the copy that I made works perfectly. I gave the original a shot and it failed with the install, switched over to the copy and all was fine. Since that has been my only snafu and it has tricked up several other systems/ drives as well, it doesn't bother me at all. I still recommend it and may be in for a third if our one other drive gets to be picky in opening.... needless to say this is a new behavior!	5
685166		2
685350	Wanted to use this to "BURN" Videos and Home Movies from My Macbook-pro Laptop to Disc!!! As that is what this was supposed to do! But Doesn't!! So it's a paper weight!! It "Plays" regular Movies but does "NOT" Burn or copy anything!!!!! Didn't come with any software!! Very disappointed!! I even bought an enclosure that hooked up to the USB to make it easy... But it didn't get it to work any better either!!	1
685342	these drives are reliable and cheap, burns cds and dvds fast, plug and play sata slot, its very quiet when reading cds	5
685420	Replaced an older SATA drive, amazon always come thru. you can always find something that fits any budget. get what you want.	5
685434	Pros: They came to me in complete working orderCons: Not much can go wrong with optical drives. Its either they work or they don't.	5
685334	Good optical drive. Recommended for anyone who needs a good cheap drive for a custom computer build. Bought as a part of a hackintosh desktop build.	5
685366	Works great...what else can I say. Since it was a bulk shipment product, it didn't come with any software...but I didn't need any, so why pay extra?	5
685404	No need for any config. Installs easy. Works easier. Does everything it says. Worth its price and then some. Recommended.	5
685393	I have been using lite-on for atleast 15 years and nvr had any complaints for them - they work great, quiet nvr had one fail me yet	5
685328	This is really good for the price. If you don't need anything more than a DVD/CD player than this is all you need. Thanks! I recommend!	5
685312	I select this DVD-RW because I have Great experience with lite-onI liked this class of BurnersI recommend this type of burners works excellent	5
685181	As promised....no problems in the installation. Install disc not needed for Linux Mint Nadia install. Works perfectly. Am planning a second build for my son. I will order another.	4
685426	Great buy. Works very well and haven't had an issues with it as of yet. Drivers were automagically found when installed on Win 7.	5
685275	Get it! Don't think about it! Just DO IT! And it's not even expensive. It was really a great buy.	5
685431	Cheapest drive I've seen. Doesn't come with any cables, but you should either already have those with you motherboard/old case. It plays/burns DVDs...what else do you expect for this price?	5
685294	Small bax inside of a big box made no sense at all, was pointless. Also got a "W" drive made in Dec 2012. Untill C4E makes W drives to be able to use BurnerMAX, stay away from this drive.	2
685415	nice cheap solution to a unit that just died. Unboxed / unwrapped and installed same day. works fine for my needs.	5

Figure 3.24: *Representatives of the summarized product selected by the proposed algorithm $\ell_{1,q}$ minimization (Elhamifar, Sapiro, and Vidal, 2012).* Some properties of the product appear in multiple different reviews.

We prove rigorously that, under the convex hull assumption , REM robustly recovers the vertices generating the data polytope. We propose an iterative algorithm to efficiently solve REM, which consists of a series of weighted $\ell_{1,\infty}$ subproblems. Finally, we show empirical evidences supporting our theoretical analysis. We show that REM is remarkably more robust than state-of-the-art vertex-identifying algorithms.

	reviewText	overall
685166		2
685450	I had to rebuild one of our computers as after a storm it wouldn't come back on... go figure since it was shut down properly. From there I had replaced most of the internals and everything was working except the DVD-RW drive was being picky--- it would work when it felt like it. I looked at the date and it was from 2006. For that I felt it served its time and I had no qualms in replacing it. Since we really do not use it except when I need to do repairs to the system or load program software that is on disc--- I was not willing to break the bank on a replacement. This one also replaced a IDE drive so no more ribbon cable which was nice as it was always a bugger to fold in the case.The only hiccup was in my system configuration. It would persist that the device was not working properly and would not recognize it properly. After a little searching I found out that in Windows 7's registry there were a possibility of four entries that could prevent this from working---- I had one of the four, deleted it, rebooted the system and Voila! it was up and running.I do not fault the drive as it was my computer/ other drive that started the whole problem. This one is definitely faster, quieter, and for the price.... it can't be beat for a DVD-RW drive. On our other computer where I do use the drive quite a bit I am tempted to replace it with one of these as well since I can also ditch the IDE cable.Update 12/2012-- I just setup a new machine and bought another one of these drives. It once again worked very well and installed without a hitch. I have one piece of software that is very particular to install--- for some reason the original disc is hit or miss, but the copy that I made works perfectly. I gave the original a shot and it failed with the install, switched over to the copy and all was fine. Since that has been my only snafu and it has tricked up several other systems/ drives as well, it doesn't bother me at all. I still recommend it and may be in for a third if our one other drive gets to be picky in opening.... needless to say this is a new behavior!	5
685420	Replaced an older SATA drive, amazon always come thru. you can always find something that fits any budget. get what you want.	5
685200	worked and still is. no problems	5
685366	Works great...what else can I say. Since it was a bulk shipment product, it didn't come with any software...but I didn't need any, so why pay extra?	5
685393	I have been using lite-on for atleast 15 years and nvr had any complaints for them - they work great, quiet nvr had one fail me yet	5
685342	these drives are reliable and cheap, burns cds and dvds fast, plug and play sata slot, its very quiet when reading cds	5
685414	Nothing much to say. Bought it so I could install W7 to my PC. Worked well still does after two years.	5
685431	Cheapest drive I've seen. Doesn't come with any cables, but you should either already have those with you motherboard/old case. It plays/burns DVDs...what else do you expect for this price?	5
685334	Good optical drive. Recommended for anyone who needs a good cheap drive for a custom computer build. Bought as a part of a hackintosh desktop build.	5
685312	I select this DVD-RW because I have Great experience with lite-onI liked this class of BurnersI recommend this type of burners works excellent	5
685407	at a good price. So stock up if you find them at a good price.I did and do I'm good for now.	5
685458	This burner definitely is a quick burner that supports writing original size and dual layerDVDs. Would recommend this unit. Used so far for about 5 mos.	5
685219	I wanted to add a faster drive to an existing computor,so I went on line to Amazon.com and Isearched for a fast drive,and I found this great deal,its really fast it works great I would recommend this product to any one thats looking to up grade there computors ability to burn CDs DVDs etc.	5
685409	It is working flawlessly, great product for your money. Haven't has any issues with it, plays anything I have used it for.	5
685426	Great buy. Works very well and haven't had an issues with it as of yet. Drivers were automagically found when installed on Win 7.	5

Figure 3.25: *Representatives of the summarized product selected by the proposed algorithm $\ell_{1,q}$ minimization (Elhamifar, Sapiro, and Vidal, 2012) with a smaller number of representatives*. The algorithm ignores the negative reviews.



	reviewText	overall
685241	Perfect performance, easy install, everything just works. Fast ejecting tray, great transfer speeds. Everything that I needed for my new PC build.	5
685342	these drives are reliable and cheap, burns cds and dvds fast, plug and play sata slot, its very quiet when reading cds	5
685420	Replaced an older SATA drive, amazon always come thru. you can always find something that fits any budget. get what you want.	5
685334	Good optical drive. Recommended for anyone who needs a good cheap drive for a custom computer build. Bought as a part of a hackintosh desktop build.	5
685231	This drive is a fantastic replacement for my dead LG IDE drive. I have used many Lite-On drives over the years since they are so reliable and was very happy to come across these. It is impossible to locate IDE DVD/CD drives these days. Checked locally and no luck. I found this SATA drive on AMAZON and figured I could work with it. Checked my MB first and found an available SATA socket. Used a 24" SATA to Right Angle SATA Serial Cable (SATA24RA1) also from AMAZON. Installation was super fast and this drive works flawlessly. Please note this is a bare bones drive (no paperwork, no software). The price on this was so good that I picked up an extra drive for future use. AMAZON shipped it very quickly and and everything was well packaged. Thank you AMAZON for the free shipping!!!	5
685350	Wanted to use this to "BURN" Videos and Home Movies from My Macbook-pro Laptop to Disc!!! As that is what this was supposed to do! But Doesn't!! So it's a paper weight!! It "Plays" regular Movies but does "NOT" Burn or copy anything!!!!! Didn't come with any software!! Very disappointed!! I even bought an enclosure that hooked up to the USB to make it easy... But it didn't get it to work any better either!!	1
685258	When your buying a CD/DVD burner for you CPU. You basically just need it to install Windows and other software. So it doesn't make much of a difference. I've had no problems at all with it for about 4 months now. The only real downside to this CD/DVD burner is the noise. It's a good bit louder than the one on my laptop. It's not annoyingly loud, so don't get me wrong. I'm just saying it could be a little more quiet.For the price I paid for this thing, which was a little under 20$. I do recommend this CD/DVD burner. Although if you have a few more dollars to spend you could always pay 10 bucks so when your doing installs it'll be a little more quiet. But I don't think many people care that much. If you need a CD/DVD player for under 20 bucks that's high quality and high speed, and don't mind a little noise during installations, this is it. It's normally quiet, it's only loud when heavy reading or writing is done.	4

Figure 3.26: *Representatives of the summarized product selected by the proposed algorithm REM*. The representatives describe distinct properties of the product. Furthermore, the algorithm is able to pick negative and not-so-positive reviews despite choosing a very compact set of representatives from a dataset with an overwhelming number of positive reviews.

References

Arora, S., R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu (2009). "A practical algorithm for topic modeling with provable guarantees". In: 28, pp. 280–288.

Chan, T.H., W.K. Ma, C.Y. Chi, and Y. Wang (2008). "A convex analysis framework for blind separation of non-negative sources". In: *IEEE Trans. on Signal Processing* 56, pp. 5120–5134.

Bioucas-Dias, J., A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot (2012). "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 5, pp. 354–379.

Kumar, A., V. Sindhwani, and P. Kambadur (2012). "Fast conical hull algorithms for near-separable non-negative matrix factorization". In: 28, pp. 231–239.

Tran, Dung N., Tao Xiong, Sang P. Chin, and Trac D. Tran (2015). "Nonnegative Matrix Factorization with Gradient Vertex Pursuit". In: *ICASSP*.

Tran, Dung N., Shuai Huang, Sang P. Chin, and Trac D. Tran (2016). "Low-rank Matrices Recovery via Entropy Function". In: *ICASSP*.

Lee, D. and S. Seung (1999). "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401, pp. 788–791.

Donoho, David and Victoria Stodden (2003). "When Does Non-Negative Matrix Factorization Give Correct Decomposition into Parts?" In:

Vavasis, S. (2009). "On the complexity of nonnegative matrix factorization". In: *SIAM J. on Optimization* 20, pp. 1364–1377.

Lee, Daniel D. and H. Sebastian Seung (2000). "Algorithms for Non-negative Matrix Factorization". In: pp. 556–562.

Arora, S., R. Ge, R. Kannan, and A. Moitra (2012). "Computing a nonnegative matrix factorization ÃŘ provably". In: pp. 145–162.

Bittorf, V., B. Recht, C. Re, and J.A. Tropp (2012). "Factoring nonnegative matrices with linear programs". In: pp. 1223–1231.

Kumar, A. and V. Sindhwani (2013). "Near-separable Non-negative Matrix Factorization with ℓ_1 and Bregman Loss Functions". In:

Gillis, N. and R. Luce (2014). "Robust Near-Separable Nonnegative Matrix Factorization Using Linear Optimization". In: *Journal of Machine Learning Research* 15, pp. 1249–1280.

Gillis, N. and S.A. Vavasis (2014). "Fast and Robust Recursive Algorithms for Separable Nonnegative Matrix Factorization". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 36, pp. 698–714.

Gillis, N. (2014). "Successive Nonnegative Projection Algorithm for Robust Nonnegative Blind Source Separation". In: *SIAM J. on Imaging Sciences* 7, pp. 1420–1450.

Esser, E., M. Moller, S. Osher, G. Sapiro, and J. Xin (2012). "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space". In: *IEEE Transactions on Image Processing* 21, pp. 3239–3252.

Elhamifar, E., G. Sapiro, and R. Vidal (2012). "See all by looking at a few: Sparse modeling for finding representative objects". In:

Cook, W.J., W.H. Cunningham, W.R. Pulleyblank, and A. Schrijver (1998). "Combinatorial Optimization". In:

Nascimento, Jose M. P. and Jose M. B. Dias (2004). "Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data". In: *IEEE TRANS. GEOSCI. REM. SENS* 43, pp. 898–910.

Qu, Q., X. Sun, N.M. Nasrabadi, and T.D. Tran (2014). "Subspace vertex pursuit for separable non-negative matrix factorization in hyperspectral unmixing". In: pp. 8115–8119.

Huang, Shuai, Dung N. Tran, and Trac D. Tran (2016). "Sparse Signal Recovery Based on Nonconvex Entropy Minimization". In: *ICIP*.

Xiao Fu, Wing-Kin Ma (2016). "Robustness Analysis of Structured Matrix Factorization via Self-Dictionary Mixed-Norm Optimization". In: *IEEE Signal Processing Letters* 23.1, pp. 60–64.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein (2011). "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine Learning* 3, pp. 1–122.

Chapter 4

Greedy Pursuit Algorithms for Separating Signals from Nonlinear Compressive Observations

The unmixing problem aims to separate a set of structured signals from their superposition. In this paper, we consider a challenging scenario in which the mixture can only be observed via nonlinear compressive measurements. In addition, the number of measurements is far less than the ambient dimension. We present a fast, robust greedy algorithm named Unmixing Matching Pursuit. We prove rigorously that the algorithm can recover the constituents from their noisy nonlinear compressive measurements with arbitrarily small error. We demonstrate the effective of the algorithm on a range of experiments, and show its superior over state-of-the-art unmixing algorithms in this context.

4.1 Introduction

It is common in practice to observe mixed signals. In a simple setup, we model a mixture $\boldsymbol{x} \in \mathbb{R}^N$ as a superposition of two unknown informative signals:

$$\boldsymbol{x} = \boldsymbol{u} + \boldsymbol{v}. \tag{4.1}$$

In the unmixing problem, one wishes to find the unknown constituents through observations of the mixed signal.

This is a challenging problem, and without any assumption, there is no hope to reliably separate the unknown component signals from their superposition. This is due to the ill-posed nature of the problem. For example, it is impossible to recover two spike signals from a mixture of form (4.1). It is thus necessary that the constituents must not look similar in order to solve the unmixing problem reliably. To formalize this assumption, one often needs to resort to specific-domain knowledge. In particular, each component signal can be assumed to be linearly expressed by a *known* set, called dictionary, of simpler objects. The elements of each such dictionary, also known as dictionary atoms, share some common structures that appear in the corresponding component signal, but are barely reflected by the other constituents. In this scenario, we say that the unknown components can be represented by some *incoherent* dictionaries (Mallat and Zhang, 1993), (Candes and Romberg, 2007), (Donoho, Elad, and Temlyakov, 2006), (Elad et al., 2005)

Instead of measuring a mixture directly, one often observes it via a set of

compressive measurements that can be acquired inexpensively:

$$\boldsymbol{z} = \boldsymbol{A}(\boldsymbol{u} + \boldsymbol{v}), \tag{4.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{m \times N}$ is a sensing matrix. The number of observations m is typical far less than the ambient dimension N, making the problem highly ill-conditioned. To make the problem solvable, the component signals should be inherently simple, in a sense that each can be well-expressed by a few atoms of the corresponding dictionary. In other words, the constituents have *sparse representation* in their dictionaries. This has received attention recently in (McCoy and Tropp, 2013), and (McCoy and Tropp, 2014).

When the constituents are sparse in some incoherent dictionaries, McCoy at. el. show that it is possible to reliably recover constituent signals from their linearly compressively observed mixture (4.2). Our proposed algorithm in this paper relies on this assumption to obtain a linear convergence rate with a relatively low sample complexity. Precise description of the assumption will be detailed in the next sections.

We consider a more general model by assuming that linear compressive samples at (4.2) are observed via a nonlinear operator $h : \mathbb{R} \to \mathbb{R}$. Furthermore, the observations can be corrupted by dense additive noise. In particular, we wish to recover constituent signals from a limited number of *noisy, nonlinear, compressive* measurements of their superposition:

$$\boldsymbol{y} = h(\boldsymbol{A}(\boldsymbol{u} + \boldsymbol{v})) + \boldsymbol{\eta}, \tag{4.3}$$

where $\boldsymbol{y} \in \mathbb{R}^m$ is the observation vector, and $\boldsymbol{\eta} \in \mathbb{R}^m$ is a random, bounded

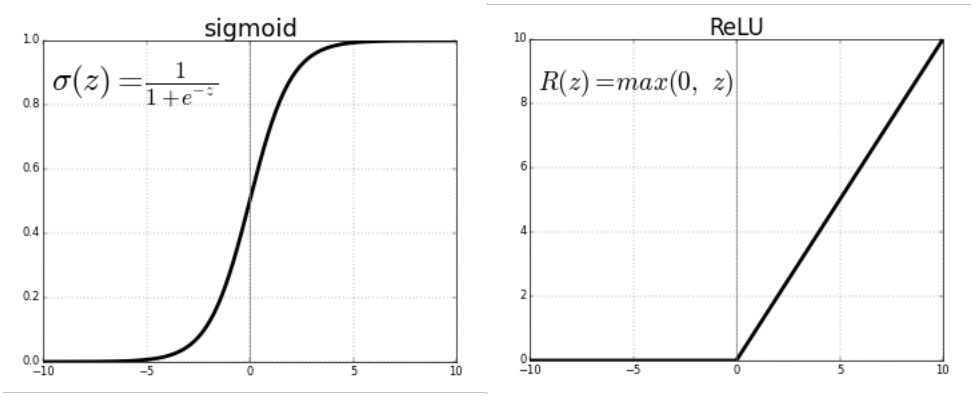


Figure 4.1: *Sigmoid and ReLU functions.*

Gaussian noise vector with zero mean. We assume that the sensing matrix $\boldsymbol{A}$, the nonlinear operator h, and the incoherent dictionaries sparsely representing the components are known. Furthermore, the signal dimension N far exceeds the number of measurements m.

In this observation model, we consider a broad set of nonlinear operators. Below is a few examples that are commonly used in neural network and signal processing literature. Their graphs are shown in Fig 4.1.

1. Rectified linear unit (ReLU) (Nair and Hinton, 2010):

$$\text{ReLU}(x) = \max(0, x)$$

2. Sigmoid:

$$h(x) = \frac{1}{1 + e^{-x}}$$

Nonlinearly observing compressive mixtures makes the unmixing problem significantly more challenging. Fig. 4.2 shows the compressive measurements of a spike and cosine mixture observed through a ReLU function. It can be seen that the entire negative part of the highly compressed observation vector

is lost. It is thus unclear that one could hope to reliably recover the component signals. Nonetheless, for certain types of mixture observation models, we rigorously show that our proposed algorithm can recover the component signals with an arbitrarily small reconstruction error from a limited number of samples.

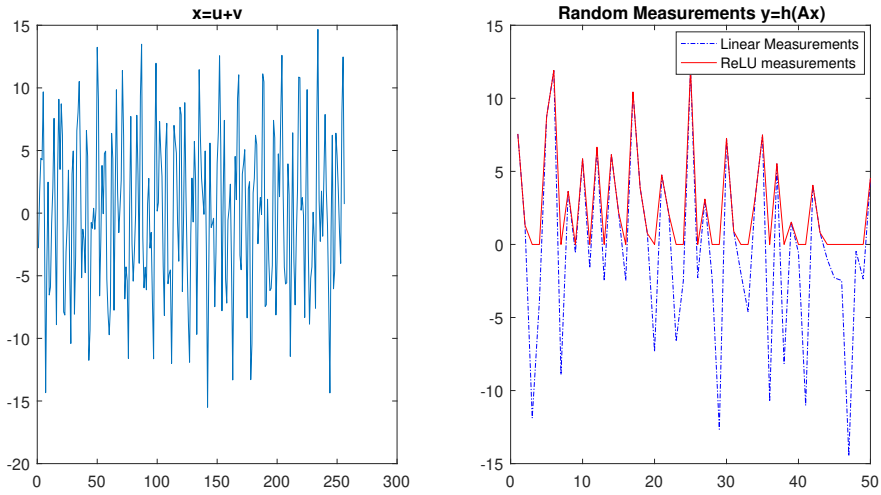


Figure 4.2: *A spike and cosine mixture signal (left) and its ReLU compressive measurements (right, in red).* The measurements contain the positive part of the mixture only.

Our contributions. We propose a fast and robust iterative algorithm called *UnmixMP* to unmix component signals under the observation model (4.3). At a high level, in each iteration of the algorithm consists of two main step. First, it aims to identify a true dictionary atom for each component signal. As we show in Section 4.3, each such atom is most correlated with the gradient of the loss function that is evaluated at the component signal estimated from the previously identified atoms. Second, we finer estimate each constituent

signal based on those chosen atoms and all corresponding dictionary atoms previously selected. *** Our algorithm is in the class of greedy pursuit algorithm which has been receive lots of attention in sparse recovery literature (Mallat and Zhang, 1993), (Tropp and Gilbert, 2007), (Dai and Milenkovic, 2009),(Needell and Tropp, 2009).

The algorithm enjoys an attractive common property of greedy pursuit algorithm that it requires no annoying parameter tuning. In its standard form, UnmixMP only requires the sparsity level of each component vectors. This information is often available from domain specific knowledge. Even when it is unavailable, one can declare successful recovery by stopping the algorithm when the reconstruction error falls below a certain small threshold. This insight is supported by our theoretical result on the upper bound of iteration number in Section 4.3.

We rigorously show that UnmixMP is fast and robust. In particular, for certain observation models, we prove that the reconstruction errors of the unmixed signals decay linearly. Unlike other convex and thresholding unmixing methods (Soltani and Hegde, 2016), this property comes at no cost of parameter tuning in the main loop of the algorithm. Furthermore, we also prove that the sample complexity to achieve this linear convergence rate is upper bounded by $\mathcal{O}\left(r \log \frac{N}{r}\right)$, where r is the total sparsity level of the component signals.

In addition, we support our theoretical analysis by various experiments on both synthetic and real image data. We demonstrate that our algorithm is significantly more robust than state-of-the-art unmixing algorithms in this

nonlinear setting.

Last but not least, each step of UnmixMP identifies atoms from constituent dictionaries separately. This allows parallelized implementation to speed up the algorithm. Detailed discussion on this subject will be presented in Section 4.4.

Applications and Related works.

The unmixing problem has been studied extensively in signal processing and statistics literature. Examples include morphological component analysis (MCA) in image processing and audio source separation (Elad et al., 2005). Another example is sparse noise correction in robust principle component analysis (RPCA) (Candès et al., 2011) and matrix completion problems (Candes and Recht, 2012) . However, our proposed algorithm is differentiated from these works. Common approaches for solving these problems assume a linear observation model in which the constituent objects are assumed to be sparse in some weakly correlated dictionaries. Furthermore, a majority of them formulates unmixing as a convex optimization problem, which are typically sensitive to parameter tuning and inferior to greedy pursuit methods in term of speed.

Perhaps the most closely related work to ours is the work by Soltani et. al. (Soltani and Hegde, 2016). In their work, the author proposed a variant of the popular iterative hard thresholding (IHT) method to unmix component signals in the nonlinear model, and achieve state-of-the-art performance. However, their algorithm requires both the knowledge of sparsity level of component signals and step-size parameter. Furthermore, as shown in the

experimental result section, our algorithm is significantly more robust than theirs in unmixing incoherent signals from ReLU and Sigmoid compressive observations.

4.2 The Unmixing Matching Pursuit Algorithm

In this section, we detail our proposed algorithm, called Unmixing Matching Pursuit (UnmixMP). We first briefly introduce the concept of sparse representation which is crucial in the development and analysis of the algorithm.

Definition 20. *The ℓ_0-norm of a vector $\boldsymbol{z}$ with respect to a dictionary $\boldsymbol{D}$ is defined as*

$$\|\boldsymbol{z}\|_{0,\boldsymbol{D}} = \inf\{r : \boldsymbol{z} = \sum_{i\in I} \boldsymbol{d}_i \alpha_i, \quad |I| = r\}. \tag{4.4}$$

In other words, $\|\boldsymbol{z}\|_{0,\boldsymbol{D}}$ is the smallest number of columns in $\boldsymbol{D}$ that can be used to linearly represent $\boldsymbol{z}$ exactly. We let $\text{supp}_{\boldsymbol{D}}(\boldsymbol{z})$ denote the index set of the atom in $\boldsymbol{D}$ constituting $\boldsymbol{z}$. Vector $\boldsymbol{z}$ is called sparse w.r.t. to the dictionary $\boldsymbol{D}$ if $\|\boldsymbol{z}\|_{0,\boldsymbol{D}}$ is relatively small comparing to the signal dimension. We are now ready to state our main assumption on the constituent signals in the nonlinear observation model (4.3).

Assumption 21. *The constituent signals $\boldsymbol{u}$ and $\boldsymbol{v}$ in the observation model (4.3) are k and s sparse w.r.t. some dictionaries $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, respectively.*

As the dictionaries are known, an appealing approach to solve the unmixing problem is thus to identify the dictionary atoms constituting the component signals. That can be done by solving the following optimization problem

with sparseness constraints (Soltani and Hegde, 2016):

$$\begin{aligned} \min_{\boldsymbol{u},\boldsymbol{v}} \quad & f(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{m}\sum_{j=1}^{m} \Gamma(\boldsymbol{a}_j^T(\boldsymbol{u}+\boldsymbol{v})) - y_j \boldsymbol{a}_j^T(\boldsymbol{u}+\boldsymbol{v}) \\ \text{s.t.} \quad & \|\boldsymbol{u}\|_{0,\boldsymbol{\Phi}} \leq k, \quad , \|\boldsymbol{v}\|_{0,\boldsymbol{\Psi}} \leq s. \end{aligned} \tag{4.5}$$

Here, the real-value function $\Gamma(\cdot)$ is defined as $\Gamma(t) = \int_{-\infty}^{t} h(z)dz$.

The reason for using the loss function in (4.5) is twofold. First of all, it can be considered as the empirical version of

$$\min_{\boldsymbol{u},\boldsymbol{v}} \mathbb{E}\left[\Gamma(\boldsymbol{a}^T(\boldsymbol{u}+\boldsymbol{v})) - y\boldsymbol{a}^T(\boldsymbol{u}+\boldsymbol{v})\right] \tag{4.6}$$

which matches the Gaussian noise assumption, as pointed out in (Soltani and Hegde, 2016). Second, in contrast to the commonly used L_2 norm whose gradient involves complicatedly computing the derivative of the nonlinear operator, its partial gradient possesses a nice, easily to compute closed form.

Lemma 22. *The partial gradients of the loss function in (4.5) is given by*

$$\nabla_u f(\boldsymbol{u},\boldsymbol{v}) = \nabla_v f(\boldsymbol{u},\boldsymbol{v}) = \frac{1}{m}\boldsymbol{A}^T\left(h(\boldsymbol{A}\boldsymbol{u}+\boldsymbol{A}\boldsymbol{v}) - \boldsymbol{y}\right) \tag{4.7}$$

The proof for this lemma is trivial, and thus is omitted. The lemma offers a useful, practical insight. Each iteration of our algorithm relies on the information encoded in each partial gradient to identify a dictionary atom for the corresponding dictionary. Lemma 22 suggests that this can be done solely based on a single proxy. This reduces computational complexity and allows parallelization to speed up the algorithm.

To solve the optimization problem (4.5), we propose a fast and robust

greedy pursuit algorithm. Each iteration of the algorithm involves first computing a proxy $\boldsymbol{g}$ which encodes useful information from previous iterations. This proxy is chosen to be the partial gradient of the objective function $f(\boldsymbol{u}, \boldsymbol{v})$ evaluated at the estimated solution from the previous iteration. As shown in Section 4.3, the proxy $\boldsymbol{g}$ is most aligned with one of the atoms in each dictionary. We thus project $\boldsymbol{g}$ onto the dictionaries, and extract an atom from each one that is most correlated to it. Finally, we estimate the unmixed components by minimizing the loss function, pretending that constituents are generated by the dictionary atoms extracted so far. This procedure is detailed in Algorithm 3. Its performance guarantee is analyzed in the next section.

Algorithm 3 Unmixing Matching Pursuit (UnmixMP)

Input: Mixture $\boldsymbol{y}$, sensing matrix $\boldsymbol{A}$, dictionaries $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$, nonlinear operator h, sparsity (k, s) or stopping criterion TOL
Initialization: $t = 0$, $\Omega_{\boldsymbol{u}}^0 = \emptyset$, $\Omega_{\boldsymbol{v}}^0 = \emptyset$
while not converged **do**

1. $\boldsymbol{g} = \frac{1}{m}\boldsymbol{A}^T \left(h(\boldsymbol{A}\boldsymbol{u}^t + \boldsymbol{A}\boldsymbol{v}^t) - \boldsymbol{y}\right)$
2. $i_{\boldsymbol{u}} = \operatorname{argmin}_l \|\mathcal{P}_{\boldsymbol{\phi}_l}\boldsymbol{g}\|_2$
   $i_{\boldsymbol{v}} = \operatorname{argmin}_l \|\mathcal{P}_{\boldsymbol{\psi}_l}\boldsymbol{g}\|_2$
3. $\Omega_{\boldsymbol{u}}^{t+1} = \Omega_{\boldsymbol{u}}^t \cup \{i_{\boldsymbol{u}}\}$
   $\Omega_{\boldsymbol{v}}^{t+1} = \Omega_{\boldsymbol{v}}^t \cup \{i_{\boldsymbol{v}}\}$
4. $(\boldsymbol{u}^{t+1}, \boldsymbol{v}^{t+1}) = \operatorname{argmin}_{\boldsymbol{u},\boldsymbol{v}} f(\boldsymbol{u}, \boldsymbol{v})$
   $$\text{s.t.} \quad \boldsymbol{u} \in \operatorname{span}(\boldsymbol{\Phi}_{\Omega_{\boldsymbol{u}}^{t+1}}),$$
   $$\boldsymbol{v} \in \operatorname{span}(\boldsymbol{\Psi}_{\Omega_{\boldsymbol{v}}^{t+1}})$$
5. $t = t + 1$

end while

Remark. Step 2 can also be called as Selection. It is akin to the selection step shared by a majority greedy pursuit algorithms in sparse recover literature. It can be inferred from Lemma 30 in the next section that each dictionary atom extracted at this step significantly reflects the structures of the corresponding

constituent.

We refer to Step 4 in the algorithm the Update step. Lemma 29 in Section 4.3 implies that the estimated components at this step look more similar to the correct constituents than the previous estimates. Intuitively, this is due to the fact that the selection step reveals more structures in the component signals.

Some important practical aspects such as convergence criteria, and how to efficiently perform the projection step and update step will be discussed in Section 4.4.

4.3 Theoretical analysis of UnmixMP

This section rigorously analyzes the performance guarantee of the UnmixMP algorithm. In particular, we first show that when the loss function $f(\boldsymbol{u}, \boldsymbol{v})$ satisfies certain restricted strong convexity (RSC) and restricted strongly smoothness (RSS) properties, the unmixed estimates converge linearly to the optimal solution of (4.5). We first introduce the following definition of RSC and RSS in the context of unmixing.

Definition 23 ((k,s)-**RSC**). *Let $\mathcal{S}_k^{\boldsymbol{u}}$ and $\mathcal{S}_s^{\boldsymbol{v}}$ be the union of all subspaces spanned by all subsets of k columns of $\boldsymbol{\Phi}$ and s columns of $\boldsymbol{\Psi}$, respectively. A function f satisfies the (k,s)-RSC property with parameter $\gamma_{k,s}^-$ if there is a positive constant $\gamma_{k,s}^-$ such that*

$$\begin{aligned} \gamma_{k,s}^- \left(\|\boldsymbol{u}' - \boldsymbol{u}\|_2^2 + \|\boldsymbol{v}' - \boldsymbol{v}\|_2^2 \right) \leq f(\boldsymbol{u}', \boldsymbol{v}') - f(\boldsymbol{u}, \boldsymbol{v}) \\ - \left\langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{u}' - \boldsymbol{u} \right\rangle - \left\langle \nabla_{\boldsymbol{v}} f(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{v}' - \boldsymbol{v} \right\rangle \end{aligned} \tag{4.8}$$

for all $\boldsymbol{u}', \boldsymbol{u} \in \mathcal{S}_k^{\boldsymbol{u}}$, and $\boldsymbol{v}', \boldsymbol{v} \in \mathcal{S}_s^{\boldsymbol{v}}$.

Definition 24 ((k,s)-**RSS**). *Let $\mathcal{S}_k^{\boldsymbol{u}}$ and $\mathcal{S}_s^{\boldsymbol{v}}$ be the union of all subspaces spanned by all subsets of k columns of $\boldsymbol{\Phi}$ and s columns of $\boldsymbol{\Psi}$, respectively. A function f satisfies the (k,s)-RSS property with parameter $\gamma_{k,s}^+$ if there is a positive constant $\gamma_{k,s}^+$ such that*

$$\begin{aligned}\gamma_{k,s}^+ \left(\|\boldsymbol{u}' - \boldsymbol{u}\|_2^2 + \|\boldsymbol{v}' - \boldsymbol{v}\|_2^2 \right) &\leq f(\boldsymbol{u}', \boldsymbol{v}') - f(\boldsymbol{u}, \boldsymbol{v}) \\ &- \left\langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{u}' - \boldsymbol{u} \right\rangle - \left\langle \nabla_{\boldsymbol{v}} f(\boldsymbol{u}, \boldsymbol{v}), \boldsymbol{v}' - \boldsymbol{v} \right\rangle, \end{aligned} \tag{4.9}$$

for all $\boldsymbol{u}', \boldsymbol{u} \in \mathcal{S}_k^{\boldsymbol{u}}$, and $\boldsymbol{v}', \boldsymbol{v} \in \mathcal{S}_s^{\boldsymbol{v}}$.

These properties play a key role in our analysis. We are now ready to state our first result.

Theorem 25 (Convergence of Algorithm 3)**.** *Suppose the loss function $f(\boldsymbol{u}, \boldsymbol{v})$ satisfies the $(2k, 2s)$-RSC and (k,s)-RSS properties with parameter $\gamma_{2k,2s}^-$ and $\gamma_{2k,2s}^+$, respectively. Let $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ be an optimal solution of (4.5). If $1 < \frac{\gamma_{2k,2s}^+}{\gamma_{2k,2s}^-} < \frac{1+\sqrt{5}}{2}$, the unmixing error satisfies*

$$\begin{aligned}&\|\boldsymbol{u}^{t+1} - \boldsymbol{u}^*\|_2 + \|\boldsymbol{v}^{t+1} - \boldsymbol{v}^*\|_2 \\ &\leq \mu(\|\boldsymbol{u}^0 - \boldsymbol{u}^*\|_2 + \|\boldsymbol{v}^0 - \boldsymbol{v}^*\|_2) + C\sigma \left(\sqrt{\frac{k}{m}} + \sqrt{\frac{s}{m}} \right), \end{aligned} \tag{4.10}$$

with a linear convergence rate $\mu = \frac{\sqrt{\gamma_{2k,2s}^+ (\gamma_{2k,2s}^+ - \gamma_{2k,2s}^-)}}{\gamma_{2k,2s}^-} < 1$, and C is a small, positive constant depending on $\gamma_{2k,2s}^-$ and $\gamma_{2k,2s}^+$.

Theorem 25 implies that when the RSS and RSC constants of the objective function f satisfies the condition stated in the theorem, the unming error decay geometrically at each iteration. Furthermore, in the noiseless case, this

implies that the unmixed estimates converge linearly to the optimal solution of (4.5). This results in the following upper bound on the number of iterations to achieve arbitrarily small reconstruction errors of the constituents.

Corollary 26. *In the noiseless setting, after*

$$t = \left\lceil \log\left(\frac{\|\boldsymbol{u}^0 - \boldsymbol{u}^*\|_2 + \|\boldsymbol{v}^0 - \boldsymbol{v}^*\|_2}{\epsilon} \right) / \log\left(\frac{1}{\mu}\right) \right\rceil \tag{4.11}$$

iterations, UnmixMP returns an unmixed solutions with accuracy ϵ.

Theorem 25 relies on certain convexity and smoothness properties of the loss function. When the derivative of the nonlinear operator h is bounded, and the dictionaries are sufficient incoherent, these properties of the loss function with a relatively low sample complexity. To state this result, we first formalize the concept of incoherent. We first define $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{\Phi} & \boldsymbol{\Psi} \end{bmatrix}$.

Definition 27. *The mutual incoherence of $\boldsymbol{D}$ is given by*

$$\mu(\boldsymbol{D}) = \sup_{i \neq j} \left| \left\langle \frac{\boldsymbol{d}_i}{\|\boldsymbol{d}_i\|_2}, \frac{\boldsymbol{d}_j}{\|\boldsymbol{d}_j\|_2} \right\rangle \right|. \tag{4.12}$$

Theorem 28 (Sample complexity)**.** *Suppose that the rows of the sensing matrix $\boldsymbol{A}$ are zero mean Gaussian vectors, the absolute value of the derivative of h is bounded within a positive interval, and $\mu(\boldsymbol{D})$ is sufficiently large. If $m = \mathcal{O}\left((s+k)\log\frac{N}{s+k}\right)$, with high probability, the loss function $f(\boldsymbol{u}, \boldsymbol{v})$ satisfies the (k, s)-RSC and (k, s)-RSS properties with parameter $\gamma^-_{2k,2s}$ and $\gamma^+_{2k,2s}$, respectively.*

The proof for Theorem 25 consists of two main steps, which guarantee that the update and the selection steps yield good constituent estimates and dictionary atoms, respectively. These two insights are summarized in Lemma 29

and Lemma 30.

Lemma 29 (Update step). *Let* $\kappa_{2k}^{\boldsymbol{u}} = \max_{|\mathcal{S}|\leq 2k} \|\mathcal{P}_{\boldsymbol{\Phi}_{\mathcal{S}}} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^*, \boldsymbol{v}^*)\|_2$ *and* $\kappa_{2s}^{\boldsymbol{v}} = \max_{|\mathcal{S}|\leq 2s} \|\mathcal{P}_{\boldsymbol{\Psi}_{\mathcal{S}}} \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^*, \boldsymbol{v}^*)\|_2$, *then*

$$\begin{aligned} &\|\boldsymbol{u}^{t+1} - \boldsymbol{u}^*\|_2 + \|\boldsymbol{v}^{t+1} - \boldsymbol{v}^*\|_2 \\ &\leq \sqrt{\frac{\gamma_{2k,2s}^-}{\gamma_{2k,2s}^+}} \left(\|\mathcal{P}_{\boldsymbol{\Phi}_{\Omega_{\boldsymbol{u}}^{t+1}}} \boldsymbol{u}^* - \boldsymbol{u}^*\|_2 + \|\mathcal{P}_{\boldsymbol{\Psi}_{\Omega_{\boldsymbol{v}}^{t+1}}} \boldsymbol{v}^* - \boldsymbol{v}^*\|_2 \right) \\ &+ C_1(\kappa_{2k}^{\boldsymbol{u}} + \kappa_{2s}^{\boldsymbol{v}}) \end{aligned} \tag{4.13}$$

where C_1 *is a small constant.*

Lemma 30 (Selection step). *Denote* $\boldsymbol{\Delta}_u = \boldsymbol{u}^t - \boldsymbol{u}^*$ *and* $\boldsymbol{\Delta}_v = \boldsymbol{v}^t - \boldsymbol{v}^*$. *Then*

$$\begin{aligned} &\|\mathcal{P}_{\boldsymbol{\phi}_{i_u}} \boldsymbol{\Delta}_u - \boldsymbol{\Delta}_u\|_2 + \|\mathcal{P}_{\boldsymbol{\psi}_{i_v}} \boldsymbol{\Delta}_v - \boldsymbol{\Delta}_v\|_2 \\ &\leq \sqrt{\frac{\gamma_{2k,2s}^+ - \gamma_{2k,2s}^-}{\gamma_{2k,2s}^-}} \left(\|\boldsymbol{\Delta}_u\|_2 + \|\boldsymbol{\Delta}_v\|_2\right) + C_2(\kappa_{2k}^{\boldsymbol{u}} + \kappa_{2s}^{\boldsymbol{v}}) \end{aligned} \tag{4.14}$$

where C_2 *is a small constant.*

With these two lemmas, we now sketch the proof of Theorem 25.

Proof of Theorem 25. Apply Lemma 29 and 30 leads to

$$\begin{aligned} &\|\boldsymbol{u}^{t+1} - \boldsymbol{u}^*\|_2 + \|\boldsymbol{v}^{t+1} - \boldsymbol{v}^*\|_2 \\ &\leq \mu \left(\|\boldsymbol{\Delta}_u\|_2 + \|\boldsymbol{\Delta}_v\|_2\right) + C' \left(\kappa_{2k}^{\boldsymbol{u}} + \kappa_{2s}^{\boldsymbol{v}}\right), \end{aligned} \tag{4.15}$$

for some small constant C'. Applying this unmixing error bound iteratively and applying Khintchine inequality to bound $\kappa_{2k}^{\boldsymbol{u}}$ and $\kappa_{2s}^{\boldsymbol{v}}$ yields (4.10). □

Proof of Lemma 29. Let $\text{supp}_{\boldsymbol{\Phi}}(\boldsymbol{u}^*) = \mathcal{T}_{\boldsymbol{u}}$ and $\text{supp}_{\boldsymbol{\Psi}}(\boldsymbol{v}^*) = \mathcal{T}_{\boldsymbol{v}}$. By the RSC and

RSS properties the objective function, and Cauchy-Schwartz inequality, we have

$$
\begin{aligned}
&\gamma^{-}_{2k,2s}\left(\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2^2+\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2^2\right)\\
&+\left\langle\nabla_{\boldsymbol{u}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\right\rangle+\left\langle\nabla_{\boldsymbol{v}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\right\rangle\\
&\leq f(\boldsymbol{u}^{t+1},\boldsymbol{v}^{t+1})-f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\\
&\leq f(\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*},\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*})-f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\\
&\leq \gamma^{+}_{2k,2s}\left(\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\|_2^2+\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\|_2^2\right)\\
&\quad+\left\langle\nabla_{\boldsymbol{u}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\right\rangle\\
&\quad+\left\langle\nabla_{\boldsymbol{v}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\right\rangle\\
&\leq \gamma^{+}_{2k,2s}\left(\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\|_2^2+\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\|_2^2\right)\\
&\quad+\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}\cup\mathcal{T}_{\boldsymbol{u}}}\nabla_{\boldsymbol{u}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\|_2\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\|_2\\
&\quad+\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}\cup\mathcal{T}_{\boldsymbol{v}}}\nabla_{\boldsymbol{v}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\|_2\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\|_2\\
&\leq \gamma^{+}_{2k,2s}\left(\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\|_2^2+\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\|_2^2\right)\\
&\quad+\kappa^{\boldsymbol{u}}_{2k}\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}-\boldsymbol{u}^{*}\|_2\\
&\quad+\kappa^{\boldsymbol{v}}_{2s}\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}}\boldsymbol{v}^{*}-\boldsymbol{v}^{*}\|_2
\end{aligned}
\tag{4.16}
$$

On the other hand,

$$
\begin{aligned}
&\gamma^{-}_{2k,2s}\left(\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2^2+\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2^2\right)\\
&+\left\langle\nabla_{\boldsymbol{u}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\right\rangle+\left\langle\nabla_{\boldsymbol{v}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*}),\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\right\rangle\\
&\geq \gamma^{-}_{2k,2s}\left(\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2^2+\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2^2\right)\\
&\quad-\|\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}\cup\mathcal{T}_{\boldsymbol{u}}}\nabla_{\boldsymbol{u}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\|_2\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2\\
&\quad-\|\mathcal{P}_{\Omega_{\boldsymbol{v}}^{t+1}\cup\mathcal{T}_{\boldsymbol{v}}}\nabla_{\boldsymbol{v}} f(\boldsymbol{u}^{*},\boldsymbol{v}^{*})\|_2\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2\\
&\geq \gamma^{-}_{2k,2s}\left(\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2^2+\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2^2\right)\\
&\quad-\kappa^{\boldsymbol{u}}_{2k}\|\boldsymbol{u}^{t+1}-\boldsymbol{u}^{*}\|_2\\
&\quad-\kappa^{\boldsymbol{v}}_{2s}\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^{*}\|_2
\end{aligned}
\tag{4.17}
$$

Here, for the sake of simplicity, with slightly abuse notations, we let $\mathcal{P}_{\Omega_{\boldsymbol{u}}^{t+1}}\boldsymbol{u}^{*}$ denote $\mathcal{P}_{\boldsymbol{\Phi}_{\Omega_{\boldsymbol{u}}^{t+1}}}\boldsymbol{u}^{*}$. Similar notations can be inferred the same way from the context. Combining (4.16) and (4.17) together with some elementary algebraic manipulations yields the desired result. □

Proof of Lemma 30. Denote $\mathrm{supp}_{\boldsymbol{\Phi}}(\boldsymbol{\Delta}_u)=\mathcal{E}_{\boldsymbol{u}}$ and $\mathrm{supp}_{\boldsymbol{\Psi}}(\boldsymbol{\Delta}_v)=\mathcal{E}_{\boldsymbol{v}}$. Using the RSC properties the objective function, and Cauchy-Schwartz inequality, we

have

$$
\begin{aligned}
& f(\boldsymbol{u}^t + \boldsymbol{\Delta}_u, \boldsymbol{v}^t + \boldsymbol{\Delta}_v) - f(\boldsymbol{u}^t, \boldsymbol{v}^t) - \gamma^-_{2k,2s} \left(\|\boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{\Delta}_v\|_2^2 \right) \\
& \geq \left\langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{\Delta}_u \right\rangle + \left\langle \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{\Delta}_v \right\rangle \\
& \geq -\|\mathcal{P}_{\mathcal{E}_u} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2 \|\boldsymbol{\Delta}_u\|_2 - \|\mathcal{P}_{\mathcal{E}_v} \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2 \|\boldsymbol{\Delta}_v\|_2 \\
& \geq -\|\mathcal{P}_{i_u} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2 \|\boldsymbol{\Delta}_u\|_2 - \|\mathcal{P}_{i_v} \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2 \|\boldsymbol{\Delta}_v\|_2
\end{aligned} \tag{4.18}
$$

Define $\boldsymbol{z_u} = -\|\boldsymbol{\Delta}_u\|_2 \frac{\mathcal{P}_{i_u} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)}{\|\mathcal{P}_{i_u} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2}$ and $\boldsymbol{z_v} = -\|\boldsymbol{\Delta}_v\|_2 \frac{\mathcal{P}_{i_v} \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)}{\|\mathcal{P}_{i_v} \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t)\|_2}$. It can then be easily shown that

$$
\begin{aligned}
& f(\boldsymbol{u}^t + \boldsymbol{\Delta}_u, \boldsymbol{v}^t + \boldsymbol{\Delta}_v) - f(\boldsymbol{u}^t, \boldsymbol{v}^t) - \gamma^-_{2k,2s} \left(\|\boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{\Delta}_v\|_2^2 \right) \\
& \geq \left\langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{z_u} \right\rangle + \left\langle \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{z_v} \right\rangle
\end{aligned} \tag{4.19}
$$

By the RSS property of the loss function, the right hand side of (4.19) can be lower bounded by

$$
\begin{aligned}
& \left\langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{z_u} \right\rangle + \left\langle \nabla_{\boldsymbol{v}} f(\boldsymbol{u}^t, \boldsymbol{v}^t), \boldsymbol{z_v} \right\rangle \\
& \geq f(\boldsymbol{u}^t + \boldsymbol{z_u}, \boldsymbol{v}^t + \boldsymbol{z_v}) - f(\boldsymbol{u}^t, \boldsymbol{v}^t) - \gamma^+_{2k,2s} \left(\|\boldsymbol{z_u}\|_2^2 + \|\boldsymbol{z_v}\|_2^2 \right)
\end{aligned} \tag{4.20}
$$

Therefore,

$$
\begin{aligned}
& (\gamma^+_{2k,2s} - \gamma^-_{2k,2s})(\|\boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{\Delta}_v\|_2^2) \\
& \geq f(\boldsymbol{u}^t + \boldsymbol{z_u}, \boldsymbol{v}^t + \boldsymbol{z_v}) - f(\boldsymbol{u}^*, \boldsymbol{v}^*).
\end{aligned} \tag{4.21}
$$

On the other hand, by the RSC property of f and Cauchy-Schwartz inequality,

$$
\begin{aligned}
& f(\boldsymbol{u}^t + \boldsymbol{z}_{\boldsymbol{u}}, \boldsymbol{v}^t + \boldsymbol{z}_v) - f(\boldsymbol{u}^*, \boldsymbol{v}^*) \\
& \geq \langle \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^*, \boldsymbol{v}^*), \boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u \rangle + \langle \nabla_v f(\boldsymbol{u}^*, \boldsymbol{v}^*), \boldsymbol{z}_v - \boldsymbol{\Delta}_v \rangle \\
& \quad + \gamma^-_{2k,2s}(\|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{z}_v - \boldsymbol{\Delta}_v\|_2^2) \\
& \geq -\|\mathcal{P}_{\mathcal{E}_{\boldsymbol{u}} \cup \{i_{\boldsymbol{u}}\}} \nabla_{\boldsymbol{u}} f(\boldsymbol{u}^*, \boldsymbol{v}^*)\|_2 \|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2 \\
& \quad - \|\mathcal{P}_{\mathcal{E}_v \cup \{i_v\}} \nabla_v f(\boldsymbol{u}^*, \boldsymbol{v}^*)\|_2 \|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2 \\
& \quad + \gamma^-_{2k,2s}(\|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{z}_v - \boldsymbol{\Delta}_v\|_2^2) \\
& \geq -\kappa^{\boldsymbol{u}}_{2k} \|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2 \\
& \quad - \kappa^{\boldsymbol{v}}_{2s} \|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2 \\
& \quad + \gamma^-_{2k,2s}(\|\boldsymbol{z}_{\boldsymbol{u}} - \boldsymbol{\Delta}_u\|_2^2 + \|\boldsymbol{z}_v - \boldsymbol{\Delta}_v\|_2^2)
\end{aligned}
\tag{4.22}
$$

Combining (4.21) and (4.22) with some simple algebraic manipulations, we arrive at the desired result. □

Proof for Theorem 28. Let $\nabla^2_{2k+2s} f(\boldsymbol{u}, \boldsymbol{v})$ be any $(2k+2s) \times (2k+2s)$ submatrix of the Hessian matrix of the loss function. As the derivative of h is bounded away from zero, it can easily be seen that in order for f achieve the desire RSS and RSC properties, it suffices to show that the minimum

and maximum eigenvalues of any $\nabla^2_{2k+2s} f(\boldsymbol{u}, \boldsymbol{v})$ is bounded within a positive interval. Similar to the proof in Soltani et. al. (Soltani and Hegde, 2016), if $m = \mathcal{O}\left((k+s)\log\frac{N}{k+s}\right)$, for some $0 < \alpha < 1$, this holds with high probability which yields the desired result. □

4.4 Practical considerations

In this section, we discuss some practical aspect of the proposed algorithm.

First of all, the selection step of UnmixMP is in fact can be written as $i_{\boldsymbol{u}} = \text{argmax}_l \, |\langle \boldsymbol{\phi}_l, \boldsymbol{g} \rangle|$ and $i_v = \text{argmax}_l \, |\langle \boldsymbol{\psi}_l, \boldsymbol{g} \rangle|$. This is the familiar form shared by a majority of greedy pursuit algorithms in sparse recovery literature.

Next, the main loop of the algorithm requires no annoying parameter tuning. In fact, UnmixMP only requires the sparsity level of each component vectors. This information is often available from domain specific knowledge. Even when it is unavailable, one can stop the algorithm when the reconstruction error falls below a certain small threshold such as 10^-3, depending on the desired speed and the unmixing errors. This insight is supported by our theoretical result on the upper bound of iteration number in Section 4.3.

Last but not least, as the gradient of the objective function has a nice closed form, the projection step can be performed efficiently using an inexpensive gradient descent method, which typically converges after a small number of iterations.

Computational Complexity. *UnmixMP* is a matching pursuit algorithm, similar to Orthogonal Matching Pursuit. If we have m measurements of an exactly $s + k$ sparse signal, the number of iterations in the of *UnmixMP* is $\mathcal{O}(s + k)$.

This is because in every iteration we recover one sparse index from the mixture components. In each iteration, we need to compute the gradient through matrix multiplications, which is $\mathcal{O}(mN)$ since we have m measurements and the signal length is N. The projection steps which actually find the magnitude of the component that we are selecting in that iteration are implemented using gradient descent. Since gradient descent runs for $\mathcal{O}\left(\frac{1}{\epsilon^2}\right)$ iterations, we get the overall computational complexity of *UnmixMP* to be $\mathcal{O}\left(\frac{mN(s+k)}{\epsilon^2}\right)$

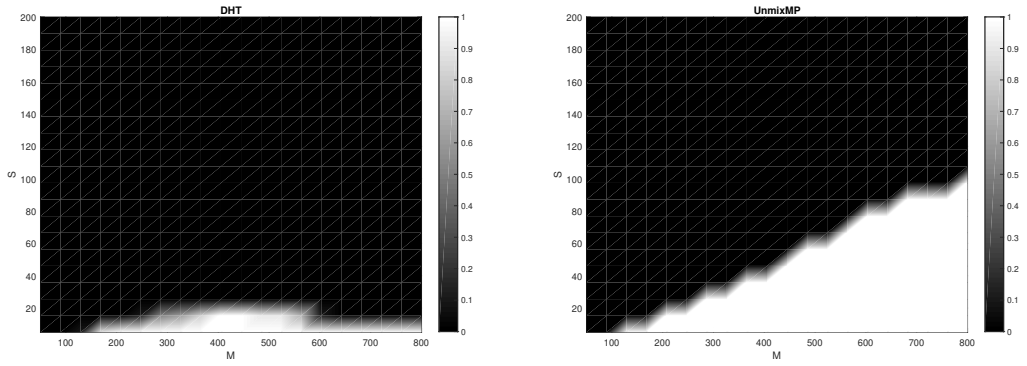
4.5 Experimental results

We perform some numerical experiments to demonstrate the effectiveness of demixMP. We compare our algorithms to the Demixing with Hard Thresholding (DHT) algorithm presented in Soltani et. al (Soltani and Hegde, 2016). We tested our algorithms on both synthetic data and real images.

First we show results from some synthetic experiments. We generate the constituent signals $\boldsymbol{u}$ $\boldsymbol{v}$ of length $N = 2^{10}$ using the Identity and Fourier bases ($\boldsymbol{\Phi}$, $\boldsymbol{\Psi}$). The measurement matrix $\boldsymbol{A}$ was chosen to be a random Gaussian matrix with normalized rows. These linear measurements were fed into a nonlinear function to generate the final measurements $\boldsymbol{y}$. We tested our algorithm with both the Sigmoid $\left(h(x) = \frac{1}{1+e^{-x}}\right)$ and ReLU $(h(x) = \max(0, x))$ nonlinearities.

The sparsity of the signals was varied from $s = 5$ to $s = 300$, and the number of measurements was varied from $m = 50$ to $m = 200$. We measured the *Cosine Similarity* between the recovered signal and original signal. We ran 10 different iterations of the experiment for each setting of m and s, and

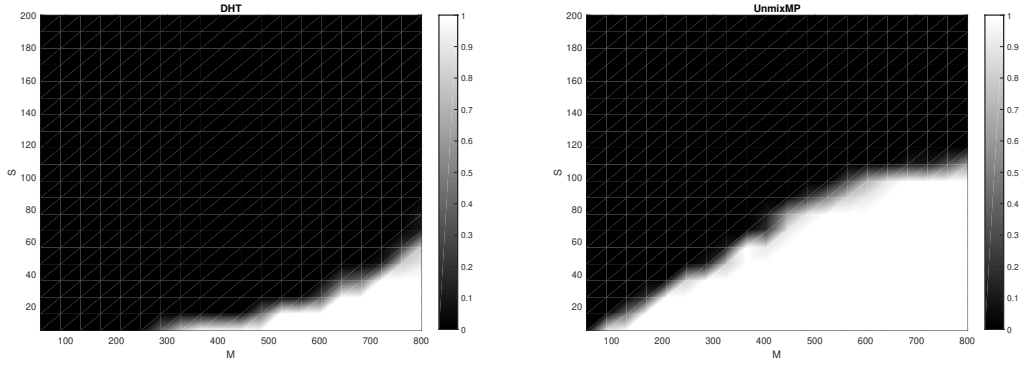
counted the number of successful recoveries of $\boldsymbol{x}$. A successful recovery was declared if the Cosine Similarity exceeded 0.95. The phase transition curves for Sigmoid and ReLU are shown in the figures 4.3 and 4.4. We observe that UnmixMP outperforms DHT in terms of better recovery at higher sparsity levels. We also see that DHT performs much worse on ReLU measurements as compared to Sigmoid measurements. Even though both DHT and UnmixMP are only guaranteed to work for smooth nonlinearities (like sigmoids), we note that UnmixMP seems to be able to handle non-differentiable functions like ReLU.



(a) DHT **(b)** UnmixMP

Figure 4.3: *Phase transition diagrams for ReLU measurements.*

For our experiments on real images we corrupted some common test images (Boats, Barbara) which were 64×64 in size, by adding a sparse (40 non-zero entries) matrix of 1s with randomly chosen support. We then tried to separate the image from the sparse noise from $m = 2000$ compressed Sigmoid measurements. We use a discrete cosine transform (DCT) matrix and an identity matrix as dictionaries for the image and noise, respectively. We compared UnmixMP to DHT in terms of PSNR of the recovered image.



(a) DHT **(b)** UnmixMP

Figure 4.4: *Phase transition diagrams for Sigmoid measurements.*

In both cases we were able to perform better than DHT. These results are reported in Table 4.1.

Image	DHT	UnmixMP
Boats	13.8 dB	**15.1 dB**
Barbara	14.1 dB	**14.9 dB**

Table 4.1: *PSNR of image recovered from Sigmoid compressive measurements.*

4.6 Conclusion

We present a greedy pursuit algorithm UnmixMP, for unmixing the components of a signal from nonlinear compressive measurements. We also prove its convergence, and give bounds on its sample complexity. We also present experiments that show the superiority of UnmixMP to other recent methods (Soltani and Hegde, 2016), especially with popular nonlinearities like Sigmoid and ReLU. We would like to explore algorithms to learn the incoherent dictionaries as well as the sparse components. We would also like to extend our

theoretical results to be able to prove convergence in the case of measurements made using non-smooth functions like ReLU.

References

Mallat, Stéphane G and Zhifeng Zhang (1993). "Matching pursuits with time-frequency dictionaries". In: *IEEE Transactions on signal processing* 41.12, pp. 3397–3415.

Candes, Emmanuel and Justin Romberg (2007). "Sparsity and incoherence in compressive sampling". In: *Inverse problems* 23.3, p. 969.

Donoho, David L, Michael Elad, and Vladimir N Temlyakov (2006). "Stable recovery of sparse overcomplete representations in the presence of noise". In: *IEEE Transactions on information theory* 52.1, pp. 6–18.

Elad, Michael, J-L Starck, Philippe Querre, and David L Donoho (2005). "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)". In: *Applied and Computational Harmonic Analysis* 19.3, pp. 340–358.

McCoy, Michael B and Joel A Tropp (2013). "The achievable performance of convex demixing". In: *arXiv preprint arXiv:1309.7478*.

McCoy, Michael B and Joel A Tropp (2014). "Sharp recovery bounds for convex demixing, with applications". In: *Foundations of Computational Mathematics* 14.3, pp. 503–567.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Tropp, Joel A and Anna C Gilbert (2007). "Signal recovery from random measurements via orthogonal matching pursuit". In: *IEEE Transactions on information theory* 53.12, pp. 4655–4666.

Dai, Wei and Olgica Milenkovic (2009). "Subspace pursuit for compressive sensing signal reconstruction". In: *IEEE Transactions on Information Theory* 55.5, pp. 2230–2249.

Needell, Deanna and Joel A Tropp (2009). "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples". In: *Applied and Computational Harmonic Analysis* 26.3, pp. 301–321.

Soltani, Mohammadreza and Chinmay Hegde (2016). "Fast algorithms for demixing sparse signals from nonlinear observations". In: *arXiv preprint arXiv:1608.01234*.

Candès, Emmanuel J, Xiaodong Li, Yi Ma, and John Wright (2011). "Robust principal component analysis?" In: *Journal of the ACM (JACM)* 58.3, p. 11.

Candes, Emmanuel and Benjamin Recht (2012). "Exact matrix completion via convex optimization". In: *Communications of the ACM* 55.6, pp. 111–119.

Chapter 5

Discussion and Conclusion

Although the curse of big data might in fact be favorable in supervised learning applications as demonstrated by recent advances in deep learning, most real-world data are unlabeled or poorly labeled. Learning from raw data in the unsupervised setting thus remains one of the most basic challenges in machine learning. As work in this area developed, an important intuition emerged: real-world data are inherently sparse in appropriate domains and can thus be approximately characterized by only a few significant features. These underlying structures not only help reveal insights from data, but can also be utilize to gain learning and inference performance in supervised learning algorithms. Problems of identifying these hidden structures from raw data thus play a particularly crucial role in machine learning in the unsupervised setting.

Among them, representative selection and mixture data unmixing serve as two of the most important and interesting problems in this category. Real-world applications, such as endmember extraction in hyperspectral imaging, justify that representative selection can sometimes be viewed as a special case

of the mixture data unmixing problem. More specifically, from the convex combination viewpoint in which each data sample can be represented as a convex mixture of the data vertices, selecting vertices as data representatives can be cast as an unmixing task whose underlying constituents are the vertices. In this thesis, we first proposed the Gradient Vertex Pursuit (GVP) and Row Entropy Minimization (REM) algorithms to efficiently identify polytope vertices of a dataset. The chosen vertices serve as extreme representatives which characterize unique properties of the data. We provided theoretical justification to confirm the correctness and robustness of GVP and REM. Strikingly, we was able to offer an optimality proof for REM, a non-convex optimization program, which is uncommon in literature. We then considered the general problem of data unmixing for under-sampled and nonlinearly observed mixture data. We proposed the Unmix Matching Pursuit (UnmixMP) algorithm which fast and robustly extracting the underlying constituting components from mixture data in this challenging setting. More precisely, we proved that UnmixMP enjoys a linear convergence rate with a relatively low sample complexity.

The fundamental idea behind the proposed algorithms is that, in a carefully constructed dictionary, a sparse representation or joint sparse representation of the input data corresponds to selecting the underlying features of interest. More specifically, in the representative selection problem, the dataset can be jointly sparsely characterized by a set of data vertices. Extracting the vertices can thus be cast as seeking a joint sparse representation of the input data in the dictionary formed by the entire dataset. This is equivalent to solving a row sparse minimization problem under the convex hull constraints.

In the unmixing problem, the constituents can be separated by finding a sparse representation of the input mixture data in a dictionary resembling these constituting components. Our proposed algorithms seek these sparse and joint sparse representation fast and robustly with rigorous performance guarantees.

Our GVP algorithm approximately solves row sparse minimization program in a greedy fashion. At each iteration, it seeks a vertex or representative from the set of unidentified vertices by minimizing a carefully chosen criterion. We proved that GVP is guaranteed to correctly extract a data vertex at each iteration. If the data polytope has s vertices, the algorithm therefore correctly identify all vertices after exactly s iterations. We empirically demonstrated the superior performance of GVP over state-of-the-art vertex extraction algorithms on both synthetic and real hyperspectral data.

After proposing GVP, we developed a new algorithm, Row Entropy Minimization (REM), to robustly solve the representative selection problem. Inspired by ideas from information theory, we first proposed a row sparsity new measure, denoted by $\|\cdot\|_{\mathrm{h},\infty}$, which is defined as the entropy function value over the set of rows of its argument. REM is a non-convex relaxation of the row sparse optimization problem which minimizes the row entropy function under the convex hull constraints. We rigorously shown that minimizing this row entropy function encourages the concentration of the rows of the coefficient matrix. In particular, a small value of the row entropy function promotes high energy rows and suppresses low energy rows of its argument. Based on this result, we established a performance bound of REM in extracting data

vertices from noisy data. More specifically, given the noisy data $\boldsymbol{Y} + \boldsymbol{N}$, where is a noisy matrix whose column norms are bounded by a positive number ϵ, REM correctly identifies all s vertices of the clean data $\boldsymbol{Y}$ provided that

$$\epsilon < \frac{\rho\gamma}{8\kappa(s+1)}. \tag{5.1}$$

Here, ρ measures how far the vertices are isolated from the non-vertex data points, γ characterizes the fatness of the data polytope, and κ bounds the norms of the data points. This strong guarantee implies that as long as the vertices are not too close to the non-vertex data points, indicated by a large value of ρ, and the polytope has a sufficiently fat shape, specified by a large value of γ, REM can robustly extract all vertices even in a significantly noisy setting. We empirically justified our theoretical results on synthetic data, and demonstrated the robust performance of REM on real video and text data. Note also that, as REM is a non-convex optimization program, rigorously providing a strong performance guarantee for this problem is alone of particular interest.

We next proposed the Unmixing Matching Pursuit (UnmixMP) algorithm to solve the unmixing problem in more general setting. Traditional unmixing methods typically assume that the considered mixture signal is a linear superposition of some hidden components. This problem is ill-posed as the number of unknowns is typically larger than the number of samples or signal dimension. We considered a more challenging setting in which the number of observed samples is far less than the signal dimension. Furthermore, these samples are indirectly observed via a nonlinear operator, such as Sigmoid or

Relu, which poses a greater difficulty to the unmixing problem. Despite these challenges, under some mild conditions on the coherence of the underlying components and the smoothness of the nonlinear operator, our proposed algorithm UnmixMP correctly separates the constituents in linear time with relatively low sample complexity. More precisely, suppose that a mixture signal linearly constituted by a few elementary components is compressively observed via a sensing matrix whose elements follow a Normal distribution. Furthermore, assume that the resulting compressive samples are observed via a nonlinear operator. UnmixMP recovers the hidden components by minimizing a carefully chosen loss function that facilitates the selection criterion at each step of the algorithm. The algorithm proceeds in greedy pursuit fashion by maintaining estimates of the supports as well as estimates of the constituents in each iteration. If the underlying constituent components are sufficiently incoherent and the nonlinear operator is sufficiently smooth, we proved that UnmixMP correctly extracts the constituents with high probability in linear time from

$$\mathcal{O}\left(r \log \frac{N}{r}\right) \tag{5.2}$$

samples. Here, r is the total sparsity of the underlying components and N is the signal dimension.

Vita

Dung Tran received his B.S. degree in Electronics and Telecommunications from Hanoi University of Science and Technology in 2009. He obtained an M.S. degree in Applied Mathematics and Statistics from Johns Hopkins University in 2017. His research interests include deep learning and classical machine learning with application to computer vision and natural language processing.